ONTOMET: Ontology Metadata Framework

A Thesis

Submitted to the Faculty

of

Drexel University

by

Luis E. Bermudez

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2004

DEDICATIONS

I dedicate this work to my wife Gaby, little Felipe,

my mom Elba, my sister Paty, and my father,

who told me to study

engineering and not music.

ACKNOWLEDGEMENTS

I gratefully acknowledge the guidance and support from all the members of my committee, Dr. Michael Piasecki, Dr. David Maidment, Dr. Hyoil Han, Dr. Claire Welty and Dr. Richard Weggel. In particular, I would like to thank my advisor Dr. Michael Piasecki, for encouraging me to pursue this stage in my life, for trusting in my ideas and my work, and for providing me with the advice whenever I needed. I would also like to acknowledge Rainer Lehfeldt, who was the first one to introduce me to the metadata and the RDF concepts; Frank Sellerohf and Christoph Lippert, for the brainstorm sessions and for the object oriented explanations that helped me to code better JAVA programs.

I would also like to thank my friends from (DICE) Drexel Informatics for Civil Engineering Research Group, Saiful, Bora, Volkan, Zafer, Yoori for their friendship, constructive discussions, and suggestions. Special thanks to Bora Beran for proofreading this thesis and Saiful Islam for walking with me inside the computer science and information systems worlds. I would like to thank in general to all the other students whom I shared moments with in the last 4 years. I would also like to mention Dr. David Lanter for his valuable suggestions and dedicated time.

I thank with my heart my lovely wife, Gaby, for encouraging me, giving me the energy, and having the required patience to support me in difficult moments. My first son Luis Felipe, was born when I was in chapter 10, almost finishing this dissertation. I thank God for this new burst of energy in my life when I needed

more than ever to complete this work. I also thank both my mother and sister in law for their advice and support.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

ABSTRACT
ONTOMET: Ontology Metadata Framework
Luis Bermudez
Michael Piasecki Supervisor, Ph.D.

Proper description of data, or metadata, is important to facilitate data sharing among Geospatial Information Communities. To avoid the production of arbitrary metadata annotations, communities agree that creating or adopting a metadata specification is needed. The specification is a document, such as the Geographic Metadata Standard (ISO 19115-2003), which provides a set of rules for the proper use of metadata elements. When a community is adopting a metadata specification it has two main concerns: 1) how can an existing specification be adopted, so that elements can be restricted and domain vocabularies be used? and 2) how can a metadata specification be mapped with another one to achieve interoperability? The two aforementioned concerns are raised due to the fact that: 1) specifications lack domain-specific elements, 2) specifications have limited extensibility, 3) specifications do not always solve semantic heterogeneities and 4) methodologies to create crosswalks among specification have not been formalized.

The main goal of this thesis is to present a feasible solution for these problems by providing a flexible environment to allow interoperations of formalized metadata specifications, extensions, crosswalks and domain vocabularies. The main contributions of this thesis are: 1) creation of an abstract model to represent metadata specifications, 2) development of a methodology to extend metadata specifications, called Dynamic Community Profile, and 3) formalization of semantic mappings to perform complex and contextual metadata crosswalks.

These three main contributions are encapsulated in a framework called Ontology-Metadata Framework or ONTOMET. ONTOMET has seven components: metadata specification, a domain vocabulary, top-domain ontology, metadata crosswalk, Dynamic Community Profile and vocabulary mapper. A Dynamic Community Profile is a metadata specification, which extends other metadata specifications and infer terms from controlled vocabularies. Vocabulary mappers solve semantic heterogeneities that appear in domain vocabularies and a metadata crosswalk expresses the semantic mappings of two specifications. Also strategies to conceptualize metadata specifications and vocabularies, are presented. Stand alone JAVA Tools and Web programs were created that implemented the methodologies presented, to allow creation of metadata instances and mappings, as well as views of hydrologic vocabularies to facilitate discovery of knowledge and resources in the Web.

CHAPTER 1: INTRODUCTION

1.1 Review of the Problem

Proper description of data is an important activity to facilitate its sharing among *Geospatial Information Communities* [18]. The information/data about data is commonly referred to as metadata. A metadata entry is composed of a *resource,* a *property* and a *value*. A *resource* can be any information medium such as a data set, jpeg image, a numerical model, the latest measurement of a station or a polyline representing a river. A *property,* sometimes called “metadata element” is a characteristic of a resource, that helps to distinguish a particular resource from others (e.g. author, date of creation, number of records etc). And a *value* is the assigned information for a resource to a property. For example, this *thesis* is a resource, that has a property, *author*, the value of which is *Luis Bermudez*.

Simply creating metadata is not enough due to the semantic heterogeneities that are present in the properties and the values. Metadata annotators may select different properties (e.g. keyword, topic, subject) or distinct values (e.g. stage, gage height, water elevation) when identifying a resource. Also, a value can have different meanings (e.g. stage, can refer to water elevation or a platform for performing arts). These semantic differences in metadata annotations hinder accurate discoveries of data and the exchange of meaningful information across communities (See Figure 1).

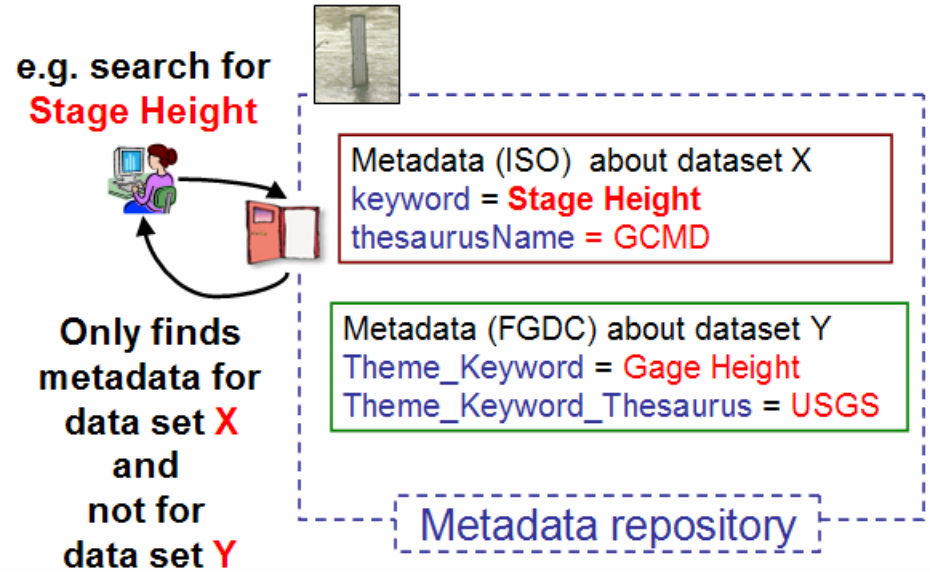


Figure 1. Discovery issues due to metadata incompatibilities

To avoid the arbitrary use of properties and values when describing a resource, rules of usage and encoding must be defined. A document that presents a set of statements which helps domain experts to formally express the rules of usage for metadata elements is called a *metadata specification.* The three major metadata specifications related to geographic digital data are*:* Geographic Metadata Standard ISO 19115:2003 [65], Dublin Core Metadata Initiative [21], and the Content Standard for Digital Geospatial Metadata, published by the Federal Geographic Data Committee [32], (These three will be referred in this thesis as ISO, FGDC and DCMI, respectively). A metadata specification is a product of a community agreement; however, it is impossible to achieve a world-wide consensus about one and only one metadata specification. Developing or adopting a metadata specification to help describe the format and the content of data to achieve interoperability has become a major issue in different earth science

initiatives (e.g. CHRONOS [15], CLEANER [17], CUAHSI [20], IRIS [63], NOKIS [88] , OASIS [91], OMG [98], MMI [82]). These initiatives have two main concerns: 1) How can a specification that already exists be easily adopted, restricting some elements and incorporating domain vocabularies?, and 2) How can a metadata specification be mapped with another one to achieve interoperability?

The two aforementioned concerns are raised due to the fact that: 1) specifications lack of domain-specific elements, 2) specifications have limited extensibility, 3) specifications do not always solve semantic heterogeneities and 4) methodologies to create crosswalks among specifications have not been formalized. In the remainder of this chapter, these problems will be discussed in detail and a strategy to overcome these difficulties will be presented.

1.1.1 Specifications' lack of domain-specific elements

When a *Geospatial Information Community* (hereafter GIC) plans to develop or use an external metadata specification, it often finds that metadata specifications are general [55] and lack of domain-specific vocabularies [26, 118]. For example, a GIC, such as a hydrologic community, will find no explicit rules in ISO, FGDC or DCMI that specify how to describe a watershed. GICs cannot determine whether the outlet location must be specified, or if the feature described needs to be called "drainage area", "watershed" or "hydrologic unit". As a consequence, communities prefer either to create their own specification, sometimes using similar elements from other specifications, such as Ecological Metadata Language EML [24]; or not using elements from any other specification, therefore producing a completely new specification, e.g. Hydrologic Markup Language HydroML [129].

The need to rewrite or create a new specification appears because of the incompatible goals among specification producers and specification implementers [53]. Producers design the specification using a top-down approach, trying to produce a general set of descriptors, while implementers seek a more detailed specification to satisfy their own needs. Normally, the producer of specifications is a publisher of standards, such as International Organization for Standardization (ISO) or the Federal Geographic Data Committee (FGDC), that sets forth a general set of descriptors. Therefore, the specifications do not provide sufficient rules for a detailed description, such as explicitly stating that the instrument number should be given when describing a gauge height measurement in a river.

1.1.2 Specifications have limited extensibility

When a GIC uses a metadata specification to fulfill its particular needs, it creates a *metadata community profile* [65]. In the creation process, a GIC can accept, discard, redefine or add elements. However, the mechanisms to perform these actions are restricted by the flexibility of the medium in which the specification is formalized. Commonly, formalizations of metadata specifications are expressed using the eXtensible Markup Language (XML) schemas [30] and the Unified Modeling Language UML [96], which are not flexible enough to redefine metadata elements.

In an XML schema, a metadata element takes the form of an XML element, which is declared using markups. For example, the following script says that an element *title* is an XML schema element and is of type string.

```
<xsd:element name=”title” type=”xsd:string”/>
```

The *title* element can be declared globally or locally and it can be a complex or a simple element. A global, a complex and a local element can all have the same name, which means that a resource can not be uniquely identified in an XML schema [56].

In a UML model it is not possible to restrict properties as it affects the membership of objects in a class and extending distributed resources in the Web will break the principle of modularization [3]. In UML, a property is an attribute of a class or an association of two classes. These are unique, and exist only if a class exists. A subclass inherits the attributes and association of its parent, but it is not possible to overwrite them. The principle of modularization states that a property belongs to one class in a defined package, but it does not accept a property belonging to two or more distributed classes.

As a consequence of the lack of flexibility of UML, specifications that are used to create a community profile, are not really extended but replicated. For example, Figure 2 depicts two metadata entities from ISO (*CI_ResponsibleParty* and *MD_Keywords*) formalized in UML. It shows possible changes in the original specification to conform to particular needs of an information community: *individualName* is mandatory, *role* should not be used, and *keyword* is restricted to a finite set of terms.

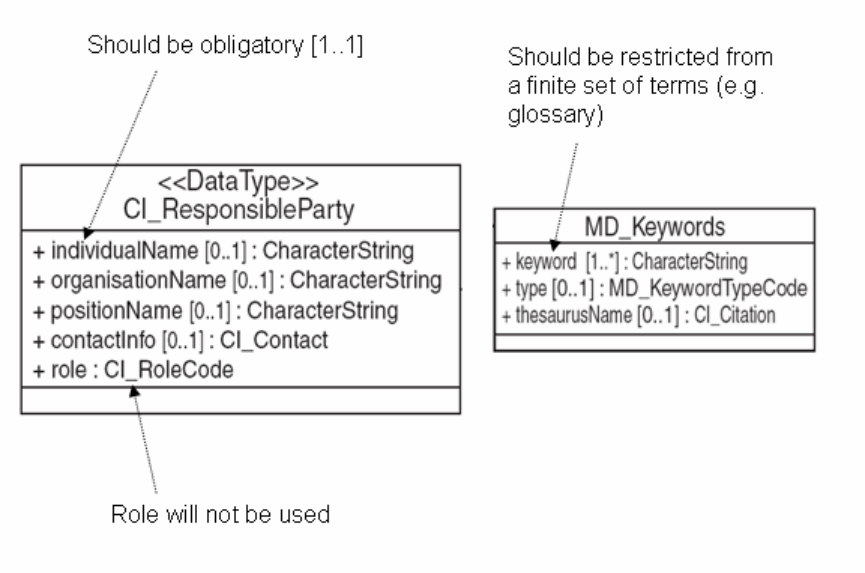


Figure 2. Example of extension of metadata specifications

Communities express these changes by rewriting the specification and not by extending the classes, mostly to preserve modularization. Figure 3 shows the classes after the changes. The implication is that the new specification will have a different namespace for its classes and attributes, and the link to the original one is lost. Also, the problem is aggravated if a GIC would like to combine elements from more than one metadata specification. At the end, GICs will have unlinked metadata specifications due to the duplication of metadata specifications, as shown in the left side of Figure 4.

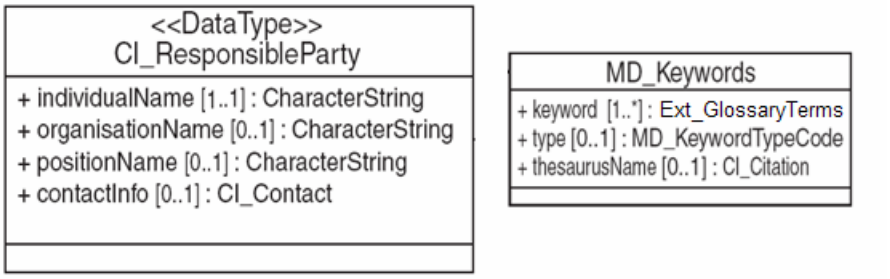


Figure 3. Classes after profile changes

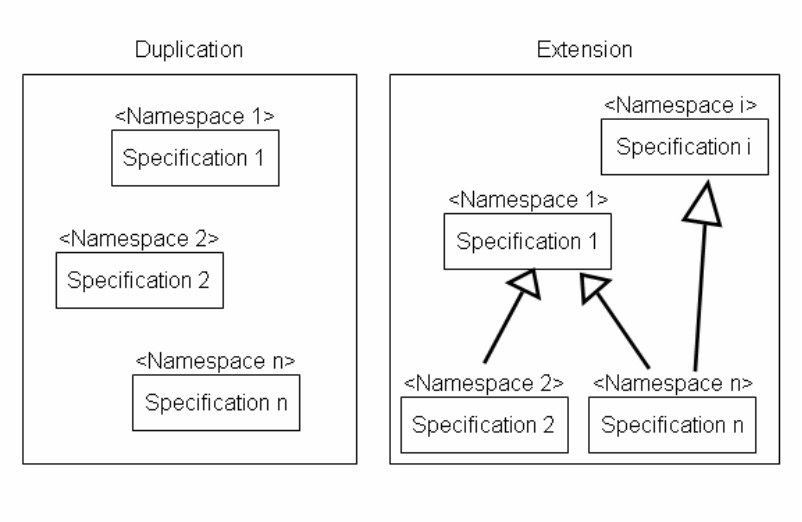


Figure 4. Community profiles: duplication vs. extension

One the other hand, if metadata specifications are extended, duplication can be avoided. They can be connected as shown in the right side of Figure 4, and, the language for expressing the metadata specification must be capable of canceling or overwriting the inherited statements. Also, elements must have a unique namespace and new elements or refinements must be able to “talk” about this uniquely identified resource.

1.1.3 Semantic heterogeneity in metadata annotations

Solely adopting a metadata specification does not solve the problem of the discrepancies of the metadata descriptions [26]. Two GICs can agree on a metadata specification, and find a way to connect their information systems to exchange data. However, semantic conflicts in metadata annotations can still appear. For example, *stage*, *gage height* and *water elevation* are different concepts that are semantically equivalent. As a consequence, the two GICs will have to resolve the semantic incompatibles to be able to exchange meaningful information. This is commonly known as the need to achieve semantic interoperability among heterogeneous systems [10, 43, 51, 52, 111]. The problem presents itself especially in *domain-specific metadata* elements (e.g. land-cover, stage height, runoff), which are metadata that capture meaningful information specific to a domain [67].

Semantic heterogeneity occurs because of the differences in world conceptualizations among individuals [36, 45, 99]. To understand the semantic heterogeneity problem it is useful to envision independent universes that surround humankind. These are the physical universe, the cognitive universe and the representation universe. The physical universe is composed of world realities such as objects and phenomena. The cognitive universe is where the perceptions or conceptualizations from the physical universe are perceived and defined, or mapped in the representation universe. The representation universe is the output of the conceptualization, composed of symbols (e.g. pictures) and formalisms (e.g. languages, conceptual models, logical models). The semantic heterogeneity problem occurs due to the differences in the mappings between the physical and

the representation universe. Discrepancies among these two universes appear because: 1) a unique world reality can have different representations (e.g. stage and gage height are different representations of water level in a section of a river); and 2) two different world realities can have a similar representation (e.g. stage can be a water level measurement or a platform for performing events).

Semantic heterogeneities are classified as [43] 1) *naming conflicts*, 2) *scaling and unit conflicts*, and 3) *confounding conflicts*. *Naming conflicts* occur when different name representations exist for a reality or when two realities have the same name representation. Different name representations that exist for a reality can be further classified as [1] synonyms, lexical variants and quasi-synonyms. Synonyms appear because of differences in linguistic origin. They exist because a reality can have a popular and a scientific representation or because of dialectic variants (e.g. stage / water elevation). Lexical variants occur because of spelling or grammatical variations (e.g. ground water / ground-water / groundwater). And, quasi-synonyms are terms that generally have different meanings, but are treated as equivalent (e.g. smoothness / roughness).

S*caling and unit conflicts* refer to two representations of one reality which are different due to the datum selected or the scales or the units of measurement. C*onfounding conflicts* occur when the representation seems to be the same, but differ in reality due to different contexts. For example, the latest reported measurement data of two different information systems can refer to different times, or a flooding stage warning may differ from river to river or agency/state policies.

In summary, semantic heterogeneities appear in metadata annotations due to the differences in the mappings between the physical and the representation

universe. Naming, scaling/units conflict, and connotation conflicts that appear in metadata annotations must be resolved to achieve semantic interoperability.

1.1.4 Informal metadata crosswalks

A crosswalk is an explicit mapping of one metadata specification to another. A National Information Standards Organization (NISO) report [115] presents the issues about creating crosswalks and states that there is not any formal methodology of specifying a crosswalk among metadata specifications. It is difficult to have a formal methodology because metadata specifications have different language representations, elements can appear in more than one context, and metadata mappings are not always a one-to-one mapping. Table 1, shows different representations for ISO and FGDC, such as Text, Document Type Definition, DTD [11], XML Schema (XSD), Unified Modeling Language (UML) or Web Ontology Language (OWL). All these formats are further discussed in section 5.4.

Table 1. Different representation of metadata specifications

	TEXT	**DTD**	**XSD**	**UML**	**OWL**
ISO			[65]	[65]	[64]
FGDC	[32]	[108]	[104]		[8]

A direct mapping from one specification to another is not always possible due to the repetition of elements in different contexts. A non-repeated element, such as *abstract* in FGDC and ISO can be mapped simply by stating <*FGDC: abstract* ***sameAs*** *ISO:abstract*>. However, not all the mappings are this obvious. For example, the element *title* in FGDC can be mapped to the element *title* in ISO,

but, a closer look at ISO, shows that t*itle* is an element of an entity called *CI_Citation*, which appears in various places or contexts in the ISO specification. *CI_Citation* can be used to describe the author of a thesaurus, or the author of a dataset. So just saying that *ISO:title* is the appropriate mapping for *FGDC:title* is not correct.

Another issue is that the mapping is not always one-to-one. One mapping example from FGDC to ISO discussed by the National States Geographic Information Council [89] is:

<FGDC:originator ***sameAS*** *ISO:Responsible Party and ISO:Role = originator>*

This mapping is a one-to-many mapping, in addition to one element having a particular value.

In summary, metadata specifications are formalized in different formats, so there is a need to represent them in a similar platform (i.e. harmonization). Also, semantic crosswalks are not always simple one-to-one mappings. Problems that occur due to contextual mappings and complex *one-to-many* or *many-to-one* mappings must be solved.

1.2 Strategy, contributions and thesis organization

The previous sections examined the problems related to extending metadata specifications and creating crosswalks. The main goal of this thesis is to present a feasible solution for these problems by providing a flexible environment to allow interoperations of formalized metadata specifications, extensions, crosswalks and domain vocabularies. The main contributions of this thesis are:

1. creation of an abstract model to represent metadata specifications;

2. development of a methodology to extend metadata specifications by creating Dynamic Community Profiles; and
3. formalization of semantic mappings to perform complex and contextual metadata crosswalks.

An abstract model for metadata specification permits visualization of all the specifications from a unique view, facilitating understanding and interoperations among metadata specifications. This abstract model presents a metadata specification as a simple tree with one root. Also a structure that allows classes and properties branched together is presented, to allow representation of conceptualized metadata specifications in a tree form.

This thesis uses the capabilities of ontologies (further explained in section 2.1) to provide a flexible environment to create metadata community profiles. Ontologies are used as a knowledge representation mechanism, for both metadata specifications and domain vocabularies. The Dynamic Community Profiles methodology uses the special characteristics of *properties* in ontologies. (e.g. properties can be restricted), to link a profile with its parent metadata specification, and to redefine inherited properties. The profile, also allows inference of terms from domain vocabularies expressed in ontologies. A community profile using this methodology was proposed for the Hydrologic Information System of the Consortium of Universities for the Advancement of Hydrologic Science Inc. (CUAHSI). The Dynamic Community Profile was tested by presenting a metadata editor named *Pangloss*. The tool is able to infer controlled vocabularies and present them in drop-down boxes. This will allow creation of metadata instances that

conform to the community profile specification. An interoperability test was performed with simple ontologies, that can serve also as an implementation guide.

To circumvent the problem of metadata crosswalks, a conceptual model based on tree-paths is proposed. The model resolves issues regarding mappings that are not one-to-one and problems related to contextual mappings. An example of semantic mappings form FGDC to ISO is presented with a tool that allows to create edit, save and load the mappings.

The previous discussed strategies are encapsulated in a framework called ONTOMET or Ontology Metadata Framework. The background concepts of ONTOMET are discussed in Chapter 2. In this chapter *ontologies*, *open and closed world assumptions*, *machine readability* and *OWL* are defined. An overview of ONTOMET is presented in Chapter 3. Chapter 4 presents traditional approaches and a comparison with ONTOMET. Chapter 5 defines and presents a survey of metadata specifications. Chapter 6 presents an abstract model for metadata specifications and a methodology to extend metadata specifications to create Dynamic Community Profiles. Chapter 7 presents the strategy to formalize and create complex semantic crosswalks on metadata specifications.

Since this thesis is based on conceptualizations of metadata specifications as well as hydrologic vocabularies, three chapters were also assigned to present conceptualization strategies for metadata specifications and vocabularies. Chapter 8 discusses the strategies to encode ISO and FGDC metadata standard. Chapter 9 presents conceptualizations of controlled vocabularies. In particular, this chapter presents a conversion into ontologies of USGS hydrologic units and Global Change Master Directory keywords. Chapter 10 discusses the creation of a top hydrologic

ontology using a top-down approach. To test the upper hydrologic ontology an inventory of terms was categorized with the proposed ontology and a Web site *Hydroogle* was created. Hydroogle allows the refinement of hydrologic searches in Google using ontological relations.

Chapter 11 presents the *Pangloss* tool, Chapter 12 discuses the creation of the Dynamic Community Profile for CUAHSI and Chapter 13 presents the interoperability test. And, Chapter 14 and 15 discusses the summary of this research and future work respectively.

CHAPTER 2: BACKGROUND CONCEPTS

This chapter presents the “ontology” concept, an overview of the Web Ontology Language (OWL), a discussion about open and closed worlds, and a definition of what is conceived as machine-readable format.

2.1 Ontologies

In computer science an ontology is an explicit and formal specification of mental abstractions, that conforms to a community agreement about a domain and design for a specific purpose [47]. It is different from the term Ontology (first letter in upper case) used in Philosophy to describe the existing things in the world [35]. Different abstractions, specifications and agreements exist among communities, so different domain ontologies exist, while only a single Ontology is possible.

An ontology provides the structure of the controlled vocabulary similar to a dictionary or a thesaurus. The vocabulary agreed to by a community is the expression of concepts (i.e. mental abstractions) of their domain. Since a concept can be expressed in different ways and differ in meaning from one person to another, the controlled vocabulary helps to solve semantic incompatibilities [10, 51, 52, 111].

A formal specification of a vocabulary can be found as a plain list of words, a dictionary, a taxonomy, an Entity-Relational (ER) diagram, an Object Model in Unified Modeling Language (UML) diagram, an eXtensible Markup Language (XML) schema and possibly many others. What makes a controlled vocabulary an ontology is that in an ontology the concepts are defined explicitly by creating classes. A class is created using a mental abstraction, which can be a

classification, an aggregation or a generalization [5]. For example, a list of terms such as *USA, Germany*, and *Colombia* do not represent any explicit conceptual relation until an explicit class *Country* is abstracted to classify them. In addition to this requirement an ontology needs to conform to strict hierarchical subclass relationships between the classes [79]. Also, classes have properties and relations among them as shown in Figure 5 .

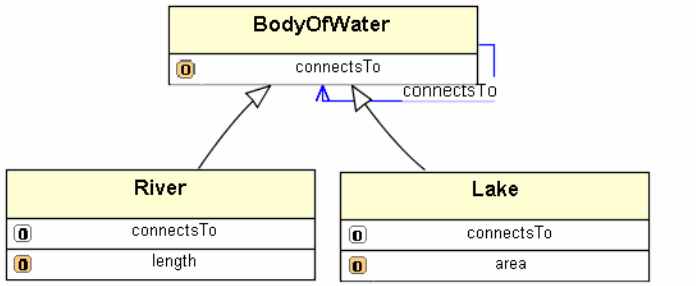


Figure 5. A small ontology example

In the small ontology example presented, the classes *BodyOfWater*, *River* and *Lake* are shown explicitly as boxes with the name of the class in bold in the first row. Properties are presented in the second and third rows. The property *connectsTo* applies to all the classes that are inherited from *BodyOfWater*, while *length* and *area* apply only to the local classes *River* and *Lake* respectively. A term such as *Delaware River* can be declared as instance (or individual) of the class River. The set of classes, properties and individuals are all part of an ontology.

Figure 5 is one of the many possible representations of an ontology. A given domain ontology should be understandable to members of a community and members of other communities, by describing it in a formal manner. For example, through the use of the Web Ontology Language (OWL), discussed in the next

section. OWL provides the mechanism to create the necessary classes and properties in a similar way as object models. Also OWL supports logical statements like inverse, transitive, symmetric and functional relations, that allows richer semantic declarations for creating control vocabulary used in metadata schemas.

2.2 Web Ontology Language

The Web Ontology Language (OWL) was selected as the vehicle to express metadata specifications because: it is a language that it is supported by the World Wide Web Consortium, W3C; it has object oriented features; it is based on a flexible graph model composed of Resource Description Framework (RDF) triples; it can be serialized in XML; and tools to interact with this language (e.g. Protégé [116]) are freely available.

OWL has components of traditional knowledge representation languages, which can be roughly classified into logical languages, frame-based languages and graph-based languages [3]. The distinction among these three can be characterized by differences in main “symbols” used to represent the knowledge. Logical languages use logical statements, such as KIF [39]. Frame-based languages use frames to represent concepts (classes), which contain slots or properties [123], such as FRL [105] and they are closely related to object-oriented languages [3, 68]. And graph-based languages use nodes and links, such as Sowa’s Conceptual Graph [114] to represent knowledge. OWL can be formalized in description logics, in an RDF graph, and in a frame-based system .

OWL is a core component of the Semantic Web [9], which is a universe of metadata and ontologies expressed in machine-readable format along with software tools that allow the understanding of semantic relations among

heterogeneous and distributed resources in the Web [25]. It is based on technologies recommended by the World Wide Web Consortium, such as the extensible Markup Language (XML), Resource Description Framework (RDF), and Uniform Resource Identifier (URI). This last one allows a Web user to display a page by clicking on a link, download a file, or to name distinctly every resource in the Web. RDF and OWL uses the URI to link, talk about, complement, use, and extend distributed resources.

RDF is based on statements that resemble simple language expressions. Statements are composed of a resource (subject) with a property (predicate) and a value (object). An example of a statement is: "*http://waterdata.usgs.gov/nwis/uv?dd_cd=01&site_no=0208758850* was created by USGS". While in the above statement only the subject is a URI, the other parts of the statement can also be represented as a URI. Figure 6 shows an excerpt in XML where the *http://purl.org/dc/elements/1.1./#creator* is also a URI. This resource, abbreviated as *dc:creator,* is an element provided by the Dublin Core Metadata Initiative [21] to describe resources in the Web.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/#">
   <rdf:Description rdf:about="http://waterdata.usgs.gov/nwis/uv?dd_cd=01&site_no=0208758850">
      <dc:creator>USGS</dc:creator>
   </rdf:Description>
 </rdf:RDF>
```

Figure 6. RDF Triple in XML

In a similar fashion geospatial data can be described by using elements that have an assigned URI and that are available as a Web resource. A community can then refer to one or more metadata standards and reuse one or many vocabularies to fit its needs. Unique identification of resources with URIs help to solve semantic heterogeneities among information communities and facilitate Web information systems to make inferences over the Semantic Web.

The framework presented in this thesis is fully interoperable with the Semantic Web, due to the fact that OWL is used as the encoding language for metadata specifications, domain vocabularies and mappings. URIs are not only used to identify resources, but also to locate them dynamically (i.e., URL). In most cases the components of the system are described using description logics, which is automatically conceived when creating ontologies conforming to OWL-Light or OWL-DL (e.i **D**escription **L**ogics). However, OWL-FULL, the third version of OWL is used in some cases when datatypes need to be restricted to a control vocabulary.

2.3 Open world and closed world assumptions

ONTOMET is built on the assumption that it is possible to reason about individuals using their own and inherited properties. The metadata specification can be expressed in a description logic system, such as OWL, that allows: 1) creation of subclasses, 2) restriction of inherited properties. Creation of subclasses guarantees that a new metadata element inherits all the properties of the super class. And, restricting a property can help to shape a metadata specification for a particular community, using similar “overwriting rules” found in object oriented systems.

OWL-Light and OWL-DL are close to description logics behavior which assumes a closed world and monotonicity. If expressions are written in OWL-Full, which is close to non-monotonic logic, some care must be taken in order to reason properly and avoid computational problems. Differences between monotonic and non-monotonic systems are explained below.

In Monotonic logic when new statements are added to the knowledge base, the conclusions increase monotonically as the knowledge base increases. It assumes a closed world where new statements will never cause previous facts to be falsified [122]. This is the case for first-order predicate logic and knowledge representation frameworks such as RDF, DAML+OIL, OWL-DL and OWL-Light [3, 58].

If an open world is assumed, new information added to the system may cause the set of conclusions to increase or to be reduced [114]; therefore, the new set of conclusions does not experience monotonic behavior. For example, if a descriptor *keyword* originally can take any string, but then is restricted to having only ten possible keywords, the new set of conclusions is reduced to ten, and the knowledge base after the last added statement is reduced. This indicates non-monotonic behavior.

Non-monotonic logic is also called defeasible inference, because earlier proofs might no longer be feasible when new information is captured. It is close to inference of everyday life, since inference can be undone by new information. Non-monotonic logic is better suited to represent inheritance systems [123] and is close to UML and Object Oriented Systems [3]. When properties are restricted, similar to

overwriting in a non-monotonic reasoning system, the more specific definitions take precedence over more general ones.

Figure 7 shows and example of a class *A'* which is a subclass of *A*. The figure on the left shows that *A'* inherits the property *p*, which has as range class *B*. The figure at the right shows *p* with a different range. The new range is class *B'*, and *B'* is a subclass of *B*. When we reason over this system and we want to get the possible range for *x*, the system will take *A'*, instead of taking *A*. While the original range of values for *p* is *B*, the figure on the right redefines the range of *p* to have only *B'* values. The ONTOMET inference system will prefer B' to B because the more specific definitions take precedence over the more general ones.

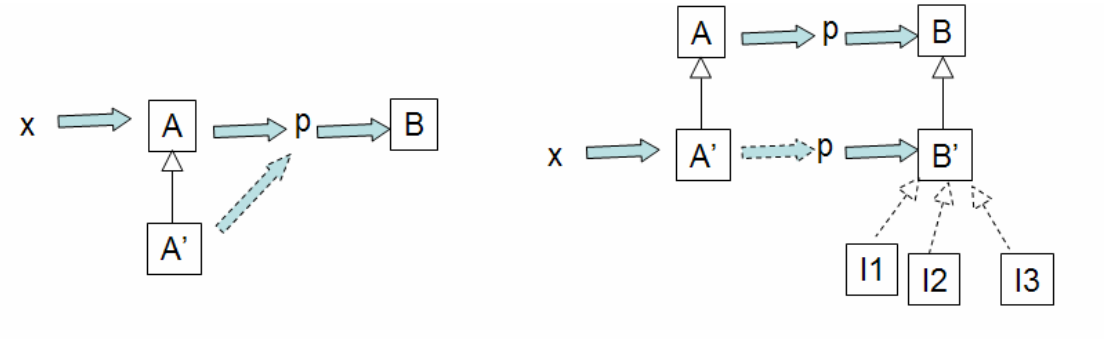


Figure 7. Reasoning over inherited properties

In this thesis, OWL is used to capture the logic of a metadata specification. ONTOMET is based on the assumption that new statements (e.g. redefining the range of an element) will never cause previous statements to be falsified. In order to "overwrite" or tell the system that the new statements (i.e. statements to create the profiles) prevail, non-monotonic inference is assumed. In ONTOMET, the restrictions defined on sub-classes will prevail among the restriction or definitions

encountered in the parent class, where more specific definitions take precedence over more general ones [123].

2.4 Machine-readable format and markup languages

A computer program can always be created to read/write information in any predefined syntax. If the syntax gains acceptance, the same program can then be used by several communities to read/write the information minimizing the time and cost of developing a diversity of software. The best example is HTML, or Hypertext Markup Language. HTML is a markup language that uses markups (“<>”) to enclose information (e.g. ** ** a paragraph ** **). Users all over the world use the HTML syntax to publish information on the Web, and browsers like Internet Explorer or Netscape are available to read this conventional syntax, to interpret the tags and to take an action (e.g. present a link an image or a list).

Information encoded in markups is commonly refer as being in *machine-readable format*, because markups are used by computer programs to understand the format. In particular, machine-readable formats are associated with encoding languages proposed by the World Wide Web Consortium (W3C). Markup languages date from late 1960’s when the Generalized Markup Language (GML) was developed. The evolution of markup languages since 1969 until today is discussed in the remainder of this section.

In 1969 IBM researchers created the Generalized Markup Language (GML) to facilitate sharing of documents. GML was used to describe the content of the document and the formatting. In 1974, Charles Goldbar, lead researcher at IBM and inspired by GML, created the Standard Generalized Markup Language (SGML) that was eventually adopted in 1986 as a standard by the International Standards

Organization (ISO 8879:1986). SGML offered a sophisticated system to create documents, which appear in the same way in any software application. However, the acceptance of SGML at the time was poor, because it was too general, overloaded with options, too complex for Web browsers to cope with and not designed for easy implementation. Web publishing needed a simplified language, easy to learn and use.

Tim Berners-Lee and Anders Berglund invented a tag-based language for marking up technical documents that can be shared over the Internet among a group of scientists in the European Organization for Nuclear Research (CERN). This language was then defined using SGML and was called Hyper Text Mark-up Language (HTML), which is the primary language for rendering Web documents today. HTML is a simple SGML type of document, with a fixed set of tags and used primarily for defining the appearance of documents in Web browsers. In the early days of the Web, HTML was a well suited language; however, Web advancements (e.g. tables, multimedia, special formats) created problems so new versions of HTML appeared. Web browsers were not always able to comply with the new versions of HTML and differences and incompatibilities started to appear in the Web. This happened mainly because HTML is not extensible; in contrast its parent SGML is fully extensible.

In 1996 the eXtensible Markup Language (XML) was proposed as an extension of SGML, enabling Web authors to fully customize their documents and enable consistent exchange of documents through the Web. XML allows information to be self-describing to the computer and easily shared [60]. The three main advantages of XML are: independence of content and presentation,

extensibility and validation. XML has been a recommendation of the World Wide Web Consortium (W3C, 2002) since February 1998, and since then it has been utilized to write existing metadata and other markup languages such as the Earth Science Markup Language (ESML, 2002), Geography Markup Language (GML. 2002) and Ecological Metadata Language (EML). Also, XML is used for encoding metadata, conceptual models and ontologies.

XML solved the syntax problem, and provided an extended platform to create other languages. However, XML schemas by themselves are not capable of expressing semantics [56]. The Resource Description Framework (RDF), then emerged to express semantics, i.e. the creation of classes and properties. Programs that rely on RDF are able to understand the tagged elements and their relations. For example, knowing that an XML element is a type of a class and that it can have properties. However, RDF does not allows restriction of inherited properties and XML datatypes are not supported. The Web Ontology Language (OWL), built on top of RDF and with XML serialization capabilities, tackles the flaws in RDF. Hence, the decision in this thesis was to use OWL as an encoded language that allows semantics restriction and XML types to be expressed in machine-readable format.

This chapter discussed the concept of ontologies, OWL, open and closed world assumptions, and machine readable format. Ontologies which are specification of conceptualizations are formalized in the Ontology Web Language (OWL). OWL provides machine readability to ontologies, and extension capabilities that allow creation of metadata profiles, by “overriding” inherited properties.

CHAPTER 3: ONTOMET OVERVIEW

The Ontology-Metadata framework (hereafter **ONTOMET**) is presented in Figure 8. The boxes are the different components of the system and the relations are presented with arrows. It depicts six different components, all of which can be formalized as an OWL ontology. A metadata specification, a domain vocabulary and a top-domain ontology are independent components and can reside in different information systems. A metadata specification can either be a metadata standard (e.g. FGDC or ISO) or a Dynamic Community Profile. A Dynamic Community Profile is a metadata specification that *extends* one or more metadata specifications and *infers* domain vocabularies that are used as controlled terms when annotating metadata. An example of a Dynamic Community Profile for CUAHSI is presented in Chapter 12. The CUAHSI profile is created by extending ISO and inferring terms from two domain vocabularies: a measurement units ontology and a science keywords ontology.

Metadata crosswalk, Dynamic Community Profile and Vocabulary Mapper, are dependent components of the independent components. A metadata crosswalk is a formalization of semantic mappings among two metadata specifications. In this thesis two possible ways to formalize a mapping are presented. One using equivalence of similar metadata tree-paths (see section 7.3) and simple mappings using OWL semantics (see Chapter 13).

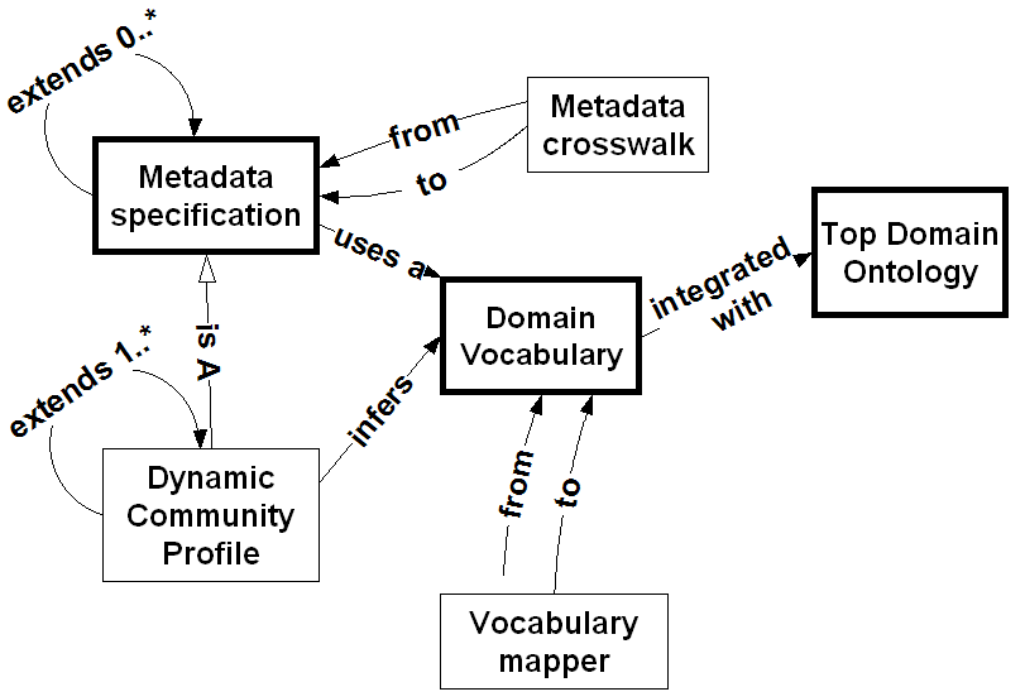


Figure 8. ONTOMET – Ontology Metadata Framework

Domain vocabularies can be merged in a top-domain ontology. This thesis presents an example of an upper hydrologic ontology (Chapter 10), where general concepts, such as measurements, observations and instruments are defined and related to help categorize hydrologic terms. Domain vocabularies are integrated to the top-domain ontology by stating that a term is a member of a concept defined in the top-domain ontology. For example, *river* is a member (or individual) of *water-body*, where *waterbody* is a class in the upper hydrologic ontology.

A main characteristic of ONTOMET is the *independence* of the components, an important feature in modern software system architectures [34]. A Dynamic Community Profile uses independent metadata specifications and independent domain vocabularies.

The methodology to create Dynamic Community Profiles overcome the difficulties of classical metadata profile approaches expressed in UML and XML schemas, which are not flexible enough to accept, discard or redefine a metadata element as well as allowing easy integration of domain vocabularies. The advantage of the methodology is that it does not require *duplication* of specifications. Each element in the metadata specification and each term from the domain vocabularies, are *uniquely identified* using URIs.

Dynamic Community Profiles methodology make use of *object-oriented* features such as inheritance to link community profiles with the metadata specification being extended. It uses the idea of controlling inheritance in non-monotonic reasoning systems, where the more specific definitions take precedence over more general ones [123].

Dynamic Community Profiles methodology make use of inferences calculated at run time dynamically. For example, keywords such as *aquifers*, *dispersion*, *water table*, can be instances of a more generic concept called “Ground Water Terms”. A logical statement can then be written saying that an element X uses all values from “Ground Water Terms”. At run time, programs can infer all the values for “Ground Water Terms”, and can then validate an instance or present a dropdown box for a user to input a valid annotation for a particular element. These inference capabilities are characteristics presented in logical knowledge based systems, and are not directly available in UML and XML schemas.

To test the framework, a suite of tools called *Pangloss* and Web applications were built to interoperate in the Semantic Web. *Pangloss* creates a run time tree with inferred vocabularies that allow creation/edition of metadata

instances based on the Dynamic Community Profile Methodology. Also with *Pangloss* explicit semantic mappings of two metadata specifications can be created. An interoperability test was coded to show all the components of the system and to present guidelines for a complete implementation.

This chapter presented an overview of ONTOMET, which is a framework that facilitates creation of Dynamic Community Profiles. ONTOMET allows more than one metadata specification to coexist and to be linked together via inheritance and formalized metadata crosswalks. Also, domain vocabularies can be inferred to provide control annotations of metadata elements and are inked together via mappers.

CHAPTER 4: TRADITIONAL APPROACHES AND COMPARISON WITH ONTOMET

4.1 Tightly and loosely coupled frameworks

The problem of metadata extensibility and crosswalks is part of a bigger problem called *information brokering* [67] whose goal is to seek interoperability among distributed and ever-expanding number of resources. A database (e.g. Oracle, MySQL), storing information about metadata instances, is considered a resource. If another database stores information also about metadata but the schema is based on a different metadata specification, then there is a need for information brokering among these two heterogeneous resources. This section provides an overview of traditional approaches and presents the differences with the strategy proposed in this thesis.

Systems to integrate heterogeneous information systems date back to early 1980's (e.g. Multibase [71]). The system was referred to as federated databases [54, 74, 112], which are also known as heterogeneous databases [107] and multi-database systems [13]. Federated databases are composed of autonomous, distributed and heterogeneous databases operating together. The first systems were tightly coupled, where a central systems administrator hardwired the mappings from a general schema to other local schemas, resolving the conflicts before they can occur (e.g. before a user queried the system).

The problem of integrating databases evolved to the need of interoperating different information systems because of the proliferation of multimedia data (e.g. images) and different database types (e.g. object-oriented vs. relational) [67]. This created loosely coupled architectures, e.g. MRDSM [73], which are based on the

premise that it is infeasible to maintain mappings from general to local schemas permanently, due to changes in locals schemas or the addition of new components to the federation. The heterogeneities in loosely coupled systems are resolved by independent modules whose messages pass through the systems via common interfaces. Widerhold [133] defines special modules, called mediators, that create information from encoded knowledge about sets or subsets of data that can be understood by higher level applications. Mediators, also called wrappers, extract data from data sets reformulating queries and can perform transformations such as those required by dates and units. They are independent from the implementation algorithm and from the different components in the federation.

Tightly coupled and loosely coupled strategies have provided integration with non-intrusive strategies but fail in the ability to scale when a large number of components are added to the system [67]. The complexity emerges due to the need for maintaining the federation updates when changes occur and when new components are added. Tightly coupled systems require a central administrator to maintain a set of functions to perform transformations from one system to another, while the loosely coupled systems, being less complex than the first one, delegates the burden to local administrators. In both cases the semantic conflicts are never resolved in an explicit fashion and are hard-coded in the conversion functions [43], worsening the problem.

This thesis presents an approach using explicit resolution of *some* conflicts before they can occur, similar to tightly coupled strategies. For this purpose, description logics and ontologies are used, based on the Web Ontology Language (OWL), instead of functions. Wrappers act on the ontologies to resolve implicit

conflicts due to the inference capabilities and the allowed logical expressions in OWL. The Dynamic Community Profiles methodology make use of *object-oriented* features, in particular inheritance and overwriting to merge metadata community profiles with metadata standards and other specifications.

For the resolution of conflicts, this thesis is based on metadata specification formalizations and not on the database schema, which can be different from the original metadata specification. OWL, an RDF based language, is used to encode metadata standards and ontologies. Tools like JENA [61], a JAVA API, presents interfaces that allows automatic storage of RDF models into relational databases, freeing the user of the burden to create database schemas that conform to the RDF schema. Converting to and from XML and relational databases is discussed in [7], and it is not the purpose of this study.

4.2 Characteristics of federated system components

Federated systems are composed of components (e.g. databases and applications) which have three characteristics [112]: distribution, autonomy, and heterogeneity. **Distribution** refers to the capability of a system to distribute data within local or federated components. An essential characteristic of ONTOMET is the use of Uniform Resource Identifiers (URIs) to identify and locate distributed resources.

A system has a design, communication and execution **autonomy** [107]. Design autonomy refers to the ability of a component to chose its own data model, information and implementation software. Communication autonomy refers to the capability of a component system to decide about what other components to communicate with, and when and what information to exchange with them. And,

execution autonomy refers to the ability of a component to execute requests independently from other federated components.

The proposed framework accepts the design autonomy of each system, meaning that any metadata specification can be used as long as it is expressed in OWL. A Communication autonomy exists, since a component of the systems (i.e. a Dynamic Community Profile) can explicitly state that it wants to use (extend) another metadata specification. The declaration of such statement is formalized in a document (XML) and not in a function of a software. The execution autonomy is left to the wrappers that use the ontology specification to perform the different actions (i.e. infer vocabularies, restrict a property, etc.).

The last characteristic of federated databases is **heterogeneity**, which occurs because of the design autonomy each component exhibits. Two types of heterogeneities can be distinguished [67]: information and system heterogeneities (See Figure 9). Information heterogeneities refer to the different ways data is organized (structural, schema), is presented (syntax and format) and is interpreted (semantic). And system heterogeneity refers to system components related to the four interoperability levels [10]: Network protocols, hardware & operating systems, spatial data files and database management systems.

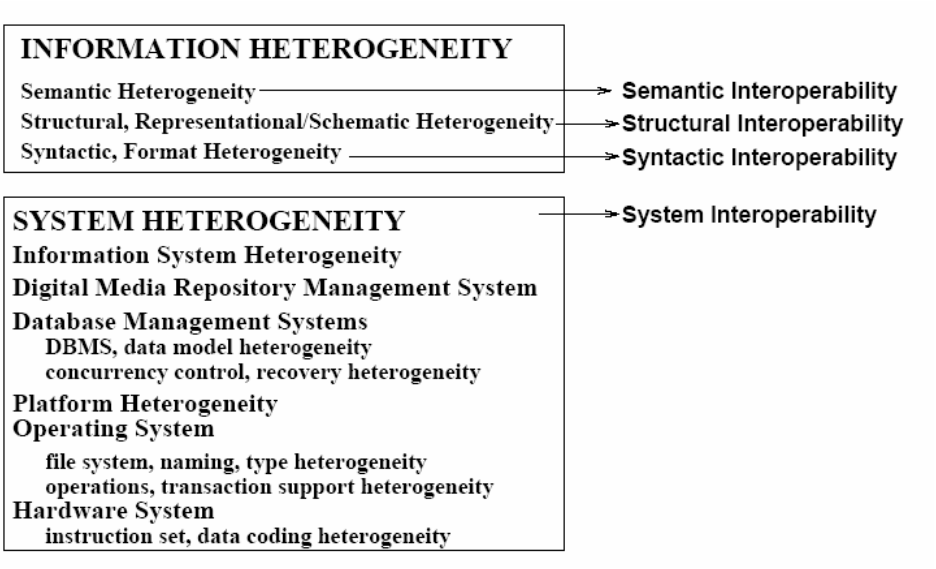


Figure 9. Taxonomy of Heterogeneity and Interoperability

Another classification is presented by Goh [43] where data conflicts due to information heterogeneities are divided in schematic, semantic and intentional conflicts. Schematic conflicts can be data type, labeling, aggregation and generalization conflicts. Semantic conflicts refer to naming, scaling and confounding conflicts and intentional conflicts can be domain and integrity constrain conflicts.

ONTOMET tackles the issue of semantic heterogeneities and metadata schema conflicts, which are similar to database schema conflicts. The conflicts are resolved explicitly and a metadata-tree-path approach is proposed to solve complex metadata specification crosswalks. The ONTOMET strategy goes beyond the federated approach. It only requires that communities express their specifications and vocabularies in a machine understandable format and make them available at a URI. The language to express specifications and vocabularies

should be one that is accepted world wide, such as OWL, proposed by the World Wide Web Consortium (W3C), .

This chapter discussed the differences and similarities between ONTOMET and traditional approaches related to information brokering. ONTOMET resolves data conflicts before they can occur, similar to tightly coupled strategies, by explicitly mapping vocabularies and crosswalking metadata specifications. ONTOMET also has the three characteristics of federated data systems: distribution, autonomy, and heterogeneity. ONTOMET can make use of distributed data by uniquely identifying with URIs every metadata element and term in a domain vocabulary. Each component in ONTOMET can exist by itself autonomously and can reside in a different server. And, OWL syntax is used to solve semantic heterogeneities among vocabularies and formalized crosswalks are used to map metadata specifications.

CHAPTER 5: METADATA SPECIFICATIONS AND ENCODINGS

5.1 Metadata definition

Metadata is data about data, “information that makes data useful” [46] . What makes data perceived as metadata is the purpose and the usage given to such data, rather than its content and structure [60]. Metadata purposes vary from organization to organization, and each one of them categorizes metadata in different ways. Indeed, a metadata element can play one or many roles. For example, the geographical coordinates of a dataset can be employed either to *discover* or to *use* a dataset. Table 2 presents a comparison of metadata categorization for three geoscience initiatives. The first column presents the purpose of the metadata as seen from a user’s point of view, and the other three columns present categorizations given by the Geochemical Earth Reference Model [40], the National Virtual Ocean Data System [90] and the Earth Science Markup Language [28]. Geographic Metadata such as ISO and FGDC are used mostly to discover and evaluate geographical data so they can be categorized as cataloging, semantic or content metadata.

Table 2. Metadata classification

User role / Initiative	GERM	NVODS	ESML
Discover	Cataloging Application	Semantic	Content
Evaluate	Application	Semantic	Content
Access	Application	Semantic	Content
Use	Application	Semantic Syntactic	Structural Semantic

5.2 Metadata specification

A metadata specification is a set of statements that help domain experts to formally express the rules of usage for metadata elements. Based on St.Pierre & LaPlant [115], in a metadata specification the following is declared and defined:

1. Declaration of an element by providing a unique identifier.
2. Declaration of a human readable label or labels in different languages.
3. Classification of an element by defining that it is a type of element category (e.g. entities in ISO), or in the case of DCMI, stating that it is a refined property (e.g. *Has Format* is sub property of *relation*.)
4. Definition of the range of an element or type of data allowed for the element. For example: string, integer, codelists or type of resource.
5. Definition for the length of characters allowed in the annotation.
6. Definition of the obligation, stating whether is mandatory, optional or conditional.
7. Definition of the occurrence: referring to the number of the times this element can appear. This is sometime refer to as cardinality. Cardinalities can be defined as minimum, maximum or with a given value.
8. Declaration of a definition.
9. Declaration of comments. This can include examples and best practices.

Using the RDF data model, a metadata specification statement can be formalized as a Subject - Predicate - Object (also known as resource-property-value triple). An example of a statement is: “abstract is a datatype property”. Figure 10 shows a formal graph, using OWL semantics.

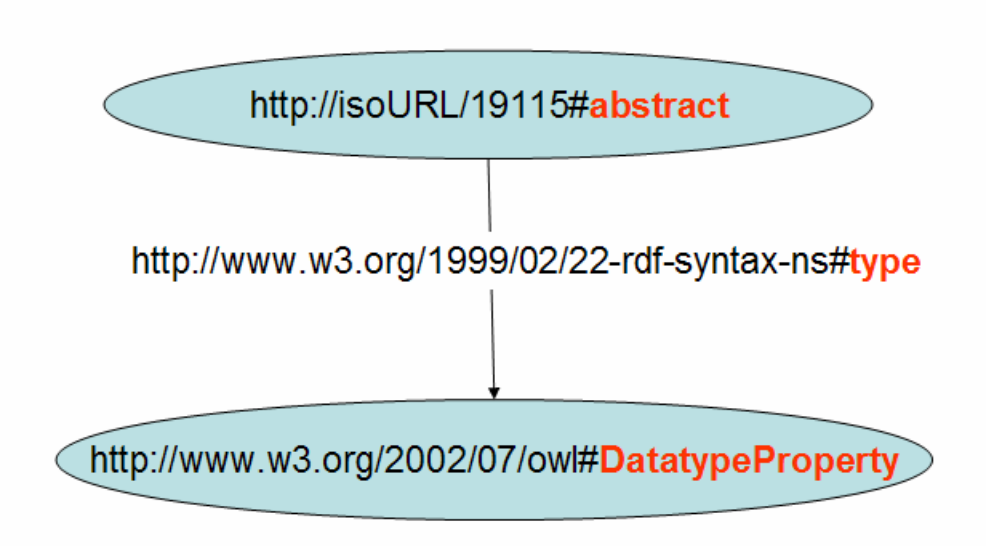


Figure 10 RDF Graph

An XML script to encode the previous triple (assuming that owl and namespaces are defined) is :

<owl:DatatypeProperty rdf:ID="abstract"/>

Using OWL it is possible to declare metadata elements as shown in Figure 10. Chapter 8 discusses the formalization of ISO and FGDC in OWL in more detail.

5.3 Related metadata specifications to hydrology

5.3.1 ISO 19115:2003

The International Organization for Standardization (ISO) is a non–governmental organization composed of a network of national standards institutes from 147 countries that coordinate and develop international standards. The ISO/TC 211 is a technical committee within ISO that deals with geographic information and is in charge of the ISO 19115:2003, the International Standard for Digital Geospatial Metadata.

ISO 19115 is presented diagrammatically in the Unified Modeling Language (UML), a graphical language promoted by the Object Management Group (OMG). The ISO 19115 Metadata set is composed of (UML) packages, aggregating in each package similar descriptors. A package is composed of entities or UML classes. For example, one package is *MD_Metadata* which encompasses the metadata entity set information.

ISO 19115 contains more than 400 elements or attributes, grouped in 95 classes, that constitute the discrete units of the metadata. Twenty two elements composed the core set that includes 7 mandatory, 4 conditional and 11 optional elements. These elements have a defined data type (integer, string, date, Class) and a domain (enumeration, or specific name of a class). For example, the element *title* has a data type of character String and the domain is free text while the element *date* is a class whose datatype is a class and whose domain is *CI_Date*. ISO 19139 is the implementation schema for ISO 19115 and as September 2004 is in its draft version which is available at the DCIWG web site [22].

5.3.2 FGDC-STD-001-1998

In the US, The National Geospatial Data Clearinghouse was established in 1994 by Executive Order 12906. Section three of this order expressed the need for developing a standardized documentation of data. This development is the FGDC-STD-001-1998, which defines the Content Standard for Digital Geospatial Metadata (CSDGM) and contains more than 200 elements.

FGDC-STD-001-1998 elements are organized into a hierarchy of compound elements. Compound elements contain other compound elements or data elements. A data element is the primitive unit and has a defined data type and

domain. The data type can be an integer, real, text, a date or time. The domain can be a list of restricted and unrestricted values, or a free value depending on the type (e.g. free data, free integer or free text).

Metadata is the top level compound element and is composed of the other 7 compound elements. The core set of FGDC-STD-001-1998 is composed of twelve elements form the two top mandatory elements: *Identification_Information* and *Metadata_Reference_Information*.

5.3.3 Dublin Core

The Dublin Core Metadata Initiative [21] publishes metadata elements for the description of resources. In 2002 it had 15 standard optional elements and by 2003 it had added more than 30 additional elements and qualifiers. The elements are presented in an XML Schema and in an RDF schema available at http://dublincore.org/documents/2002/05/15/dcq-rdf-xml/#DCQS. DCMI defines properties but does not define the ranges and elements of these properties, thus provides great flexibility for its use. There are some defined classes to help identify schemas that maybe used in the description. For example, the subject schemes supported by DCMI are: LCSH, Mesh, DDC, LCC, UDC .

The success of Dublin Core is its simplicity of 15 non-mandatory elements and its additional small set of qualified elements. Dublin Core properties and RDF properties are linked by inheritance, using the rdf:subproperty relationship.

5.3.4 Directory Interchange Format

The Directory Interchange Format, DIF [84], is a collection of fields that store information about data. It was a consequence of the Earth Science and

Applications Data Systems Workshop (ESADS) on catalog interoperability (CI) held in February in 1987. DIF was approved in 1988.

In DIF, eight fields are mandatory while the others are optional. The values that each field takes is either text or a value from a controlled vocabulary. Some fields are group fields, that can contain other fields. For each element, the length of the field is provided as a maximum number of characters.

5.3.5 Earth Science Markup Language

The Information Technology & Systems Center of the University of Alabama is developing the Earth Science Markup Language [28] which makes possible sharing and transformation of files from one format to another. ESML defines documents in XML to describe any format file with the required elements so a computer program can interpret the data. Figure 11 presents an ESML example describing a binary file with 2 header lines and an array of values with size X and size Y.

ESML provides a way to describe the format of a file but suffers from a lack of *content metadata* descriptors that metadata standards provide, such as FGCD and ISO (e.g. quality descriptors).

5.3.6 Geography Markup Language

GML is an XML based encoding standard for geographic information developed by the OpenGIS Consortium [19]. GML is well developed to describe geometries and geographical relations. The GML specification starts with an abstract feature model using an object oriented approach and ends with a set of XML schema specifications. An important characteristic of GML is that it provides a

set of XML schemas that may be used to construct an application schema. The application schema declares the feature and property types of interest of a particular domain. GML is also known as ISO 19136.

```
<a:ESML
        xmlns:a="ESML"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="ESMLR:\Schema\ESML.xsd">
    <SyntacticMetaData>
        <Binary>
            <Structure instances="1">
                <Header type="Int32" name="sizeX" symbol="true"/>
                <Header type="Int32" name="sizeY" symbol="true"/>
                <Array occurs="$sizeX">
                    <Array occurs="$sizeY">
                        <Field name="BrightnessTemp" type="Int32" order="LittleEndian"/>
                    </Array>
                </Array>
            </Structure>
        </Binary>
    </SyntacticMetaData>
</a:ESML>
```

Figure 11. Example of an ESML file

5.3.7 Ecological Metadata Language

EML [24] is an open source project that provides a metadata specification, based on XML schemas, to guide the documentation and sharing of ecological data. It uses similar metadata elements found in other standards, such as ISO, FGDC, DCMI, the biological profile of the Content Standard for Digital Geospatial Metadata, the OpenGIS Consortium's Geography Markup Language (GML), the Scientific, Technical, and Medical Markup Language (STMML), and the Extensible Scientific Interchange Language (XSIL).

EML differentiates four type of resources: dataset, literature, protocol and software resources. EML resources provide the general descriptors of all the resources (e.g. creator and title), and each resource has its own special descriptor

specific to its domain (e.g. ISBN only applies to literature resources). EML, also provides a set of modules to be used in the description of a resource, such as :

- EML methods module: Methodology used to collect the data.
- EML project module: Research context and design of the experiment.
- EML access module: Rules for accessing the data and the metadata.
- EML entity module: Logical structure of the dataset.
- EML physical module: Characteristics of the file.
- EML party module: Describes the originator of the resource.
- EML coverage module:
    - Geographic: Bounding coordinates or G-Ring polygon (order list of latitude longitude coordinates and inner polygons that excludes the region.
    - Temporal: Refers to ISO 8601.
    - Taxonomical: hierarchy of rank names, values and common names (e.g. Rank Name = Kingdom, Rank Value = Animalia, Common Name = Animals, and so on down the hierarchy.)

5.3.8 ADN Framework

An important activity that facilitates description of resources that can be used in learning education is ADN or ADEPT-DLESE-NASA metadata framework [23]. ADN, which is presented in XML schemas, identifies required metadata for a cataloger provider and for a collection provider. The former is composed of 11 elements: title, URL, Description, subject, technical requirements, recourse type, audience, copyright, cost, resource creator and resource cataloguer. And the

required metadata for a collection builder provider is: language of the resource, language of the metadata, copyright of the metadata, terms of use, metadata framework, creation date, accession date, catalog name and number and record status. All the other metadata elements are define as robust metadata, which includes coverage, science standards, geography standards, relationship between resources, keywords, etc.

5.3.9 ANZLIC

The Australian Spatial Data Infrastructure (ASDI) is a framework that supports the usage of spatial information linking users and data providers of spatial information. One of its objectives is to provide spatial metadata and standards. ANZLIC is the geographic metadata standard for New Zealand and Australia. It presents an annex with mappings to ISO-19115 metadata.

5.4 Metadata specifications formalizations

This section presents the most common formalizations for the metadata specifications previously described and discusses some differences among these formalizations. The possible formalizations of metadata specifications, as well as their strengths and weaknesses are as follows:

- ASCII text: Lacks machine readability.
- DTD Document Type Definition [11]: Defines legal building blocks of XML documents. Lacks of support for both cardinality and datatypes.
- XSD (XML Schemas [30]): Has the function of a DTDs and provides support for cardinality and datatypes. URI uniqueness is not guaranteed due to name conflicts in global and local elements.

- UML Unified Modeling Language [96]: A language to create conceptual models. Allows to create classes, attributes and associations among classes. It is not directly expressed in machine-readable format; however programs to edit UML, can export the model to XML Metadata Interchange Format or XMI [97]. XMI is a special XML schema that allows expression of UML models. UML has also limitations on the extensibility and using URIs, to identify uniquely distributed resources is not supported.
- RDFS Resource Description Framework Schemas [12]: Allows expression of semantics like class and properties, and can be serialized in XML. Cardinalities and property restrictions are not supported. URI uniqueness is guaranteed.
- OWL Web Ontology Language [6]: Built on top of RDF, supports restrictions, cardinalities and XML datatypes.

Table 3 shows formalizations for different types of metadata specifications related to digital libraries and geospatial data. It is important to notice that any specification can be presented in different ways. For example, FGDC is presented in text, DTD, XSD and OWL.

A metadata specification can be presented either in a informal or in a formal manner. An informal approach is one that specifies a metadata specification in plain ASCII text. A formal approach, on the other hand, is one that expresses a metadata specification as a conceptual model in terms of classes and properties, or as an application schema. The two most common conceptual model representations are UML and XMI. A metadata specification can also be expressed

as an application schema that helps applications to interact with it. Application schemas are in machine-readable format, commonly represented in XML.

Table 3. Formalization of metadata specifications

	TEXT	**DTD**	**XSD**	**UML**	**RDFS**	**OWL**
ISO			[65]	[65]		[64]
FGDC	[32]	[108]	[104]			[8]
GML			[93]	[93]		
DCMI					[21]	
EML			[24]			
DIF	[83]	[83]				
HYDROML			[129]			
ANZLIC	[2]	[2]				
ADN			[23]			

The Web Ontology Language, OWL [6] is a recent technology recommended by W3C that is capable of representing sophisticated conceptual models and at the same time serve as an application schema. Differences and similarities between XML schemas, RDF and ontology languages are well documented in the literature [42, 56, 62]. Differences between UML and DAML+OIL, which is very similar to OWL, can be found in [3]. And a mechanism to interoperate from application conceptual schemas to ontologies can be found in [65].

The main advantages of OWL over specifications in plain ASCII text, XML schemas, and UML can be summarized as follows: 1) OWL is able to represent

conceptual models with classes and properties and their relationships, similar to UML, which is not possible in XML schemas, 2) OWL is expressed in machine-readable format using RDF/XML which plain ASCII text and UML are not, 3) OWL is built on top of the Resource Description Framework (RDF) model, where a resource is expressed as a Uniform Resource Identifier (URI) that is not part of the UML and XML schema model [56]. Retrieving elements from an XML schema may cause difficulties because complex types, global and local elements, can have the same name, and 4) OWL allows to restrict inherited properties, a feature that is not present in UML. Figure 12, shows the different implementations for metadata specifications organized by level of conceptualization, machine readability and extensibility. Extensibility is defined as the possibility of extending distributed Web resources. To talk about distributed resources XML make use of additional technologies like XML Linking language [131] and the XML Path Language [132]. Figure 12 also presents XMI (XML Metadata Interchange), which is a special XML schema that allows expression of UML models.

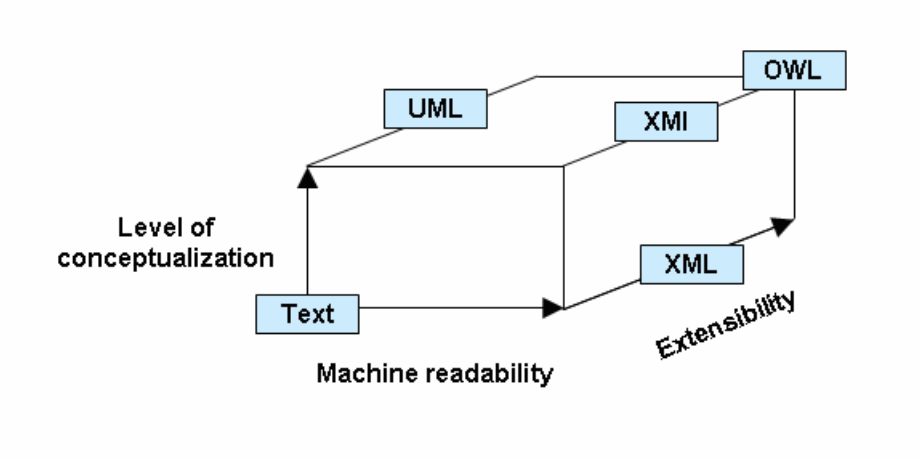


Figure 12. Metadata specifications dimensions

The advantage of UML over OWL is that UML is a mature language and has become the standard tool to create conceptual models. However, very recently applications have become available to create OWL ontologies like Protégé (http://protege.stanford.edu/) and others listed in http://Web.daml.org/tools/. Also, some research efforts are underway that focus on developing UML tools to create ontologies [3]. To create the specifications and controlled vocabularies, Protégé in combination with the plug-in tool ezOWL (http://iWeb.etri.re.kr/ezowl/index.html), that allows the visualization of models is used. To create the community profiles two tools were used. Protégé to create simple restrictions and *Pangloss* to create more sophisticated statements, such as the core paths.

This chapter defines a metadata specification and presents a survey of metadata specifications related to hydrology. It also, presents the strengths and weaknesses of the different types of encodings for metadata specifications.

CHAPTER 6: DYNAMIC COMMUNITY PROFILES

6.1 Definition

A Dynamic Community Profile is a metadata specification, formalized in a machine-readable format, in which configurable extension declarations are allowed to meet the needs of a particular community. The following extension declarations are allowed in a profile:

- Accept an element (set as core).
- Discard an element (set as not core).
- Set an element as mandatory.
- Set an element as optional.
- Set to allow multiple values.
- Set to not allow multiple values.
- Change the range of possible values .
- Add a new element.
- Change the name or label and the description.
- Set the length of the field.

A Dynamic Community Profile is a metadata specification that *extends* one or more metadata specifications and *infers* domain vocabularies that are used as controlled terms when annotating metadata. A conceptual view of a Dynamic Community Profile is depicted in Figure 13. It is composed of an extended metadata specification, a domain vocabulary and tree-paths, which is further explain in chapter 6.2. This can be thought of a framework consisting of three different ontologies, each of them residing in a different file. By using tree-paths it is possible to uniquely identify repeated metadata elements in a metadata specification. More about the tree-paths is discussed in chapter 6.2.

A metadata specification uses a domain vocabulary (e.g. code list in ISO), but an extended metadata specification can make use of a domain vocabulary (i.e. terms that specifically pertain to a community), by using the inference capabilities built in ontologies in OWL. The profile is dynamic because some of the extended declarations are realized at run time. All the terms in the vocabulary might not be known when declaring a profile. The terms are known at run time, when the inference algorithm provides the terms inferred.

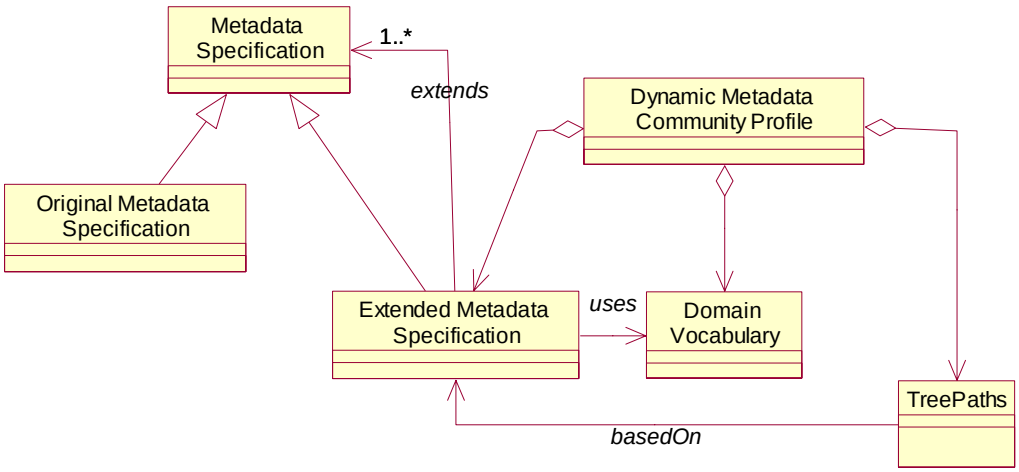


Figure 13. Dynamic Community Profile

The extended metadata specification must be interoperable with the original specification that has been extended. This implies that it should be possible to find a metadata instance created from the extended metadata, while using queries from the original metadata specification. It may be necessary to apply some type of query language which accepts rules to inference the instances from the profile, such as RDQL [109]. Chapter 13 presents a test where instances from Dynamic

Community Profiles are interoperable with the original specification. It also shows examples of RDQL statements in the context of metadata specifications.

Before going into the detail explanation of the methodology to create Dynamic Community Profiles, an abstract model for metadata specifications is presented in the following section.

6.2 Abstract model for metadata specifications

To refer to metadata items used in any metadata specification, the term *metadata descriptor* will be used in this chapter to avoid conflicts between the terminology used by different specifications. A metadata descriptor is a metadata element when we refer to FGDC, entity or element when we refer to ISO, class or attribute (or association) when we refer to UML metadata models, a property when we refer to DCMI and fields when we refer to DIF.

Based on similarities of metadata model specifications and the definition discussed in 5.2 an abstract model is presented in Figure 14. A metadata specification has a root which is a *ComplexDescriptor*, which is a type of *Descriptor*. A *Descriptor* has the following properties: *label*, *definition*, *isMandatory*, and *isRepeatable*. A descriptor can be a *ComplexDescriptor* or a *SimpleDescriptor*. A *ComplexDescriptor* contains one or more *Descriptor*s, that can either be *ComplexDescriptor*s or *SimpleDescriptors*. A *SimpleDescriptor*, is one that does not contains other descriptors and has a range of values, specified by a datatype or a controlled vocabulary.

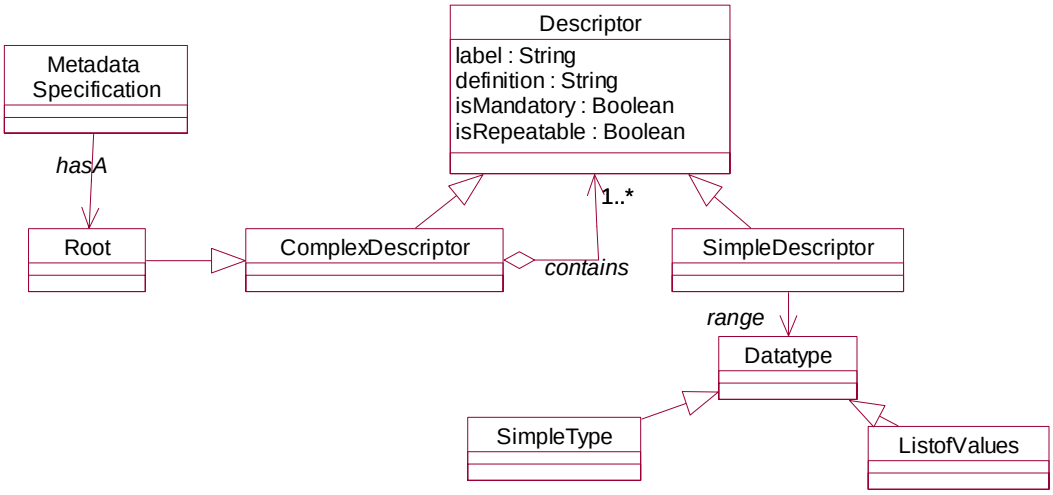


Figure 14. Abstract Model for Metadata Specifications

A metadata descriptor can be identified as a sequence of labels, starting from the root label. Every metadata specification has a root descriptor. Root descriptors for ISO, FGDC, DCMI and DIF are *MD_Metadata*, *Metadata*, *DC* and *DIF* respectively. A root is a *ComplexDescriptor*, which can contain other descriptors, until it reaches a *SimpleDescriptor*. Table 4 presents examples of how the "publishing date" descriptor is depicted in different specifications.

The ISO 19115:2003 metadata specification uses the format presented in Table 4, when referring to core elements (see table 3 of ISO specification). The symbol ">" denotes relation to other classes, while the "." denotes a class-attribute relation. This wording format describes exactly a particular descriptor and its position in the specification. In the case for ISO, this is very important because some descriptors can be use in several parts of the specification. For example, ***CI_Citation*** is used in ***MD_Keywords.thesaurusName. CI_Citation*** and in ***MD_DataIdentification. citation.CI_Citation.***

Table 4. Metadata descriptors sequence.

ISO	***(MD_Metadata > MD_DataIdentification.citation > CI_Citation.date)***
DCMI	DC.date
FGDC	Identification-Information_Citation_Citation-Information_Publication-Date (implicit root = Metadata)
DIF	DIF Data_Set_Citation Dataset_Release_Date

The encoding of DCMI shown in Table 4 is used when annotating metadata in HTML pages. FGDC elements are depicted in the hierarchical form shown in Table 4 when encoded in Metadata Interchange Format [55] in: http://cuahsi.sdsc.edu/html_source/_metadata_resources.html. And DIF can be presented as in Table 4, by creating an implicit root name DIF.

The previous examples have a hierarchical tree-like data structure. A tree has a root, or parent node. A root has children. A child can be a branch or a leaf. Branches are *ComplexDescriptors* and the leaves are *SimpleDescriptors*. Following the pattern: ***Root.ComplexDescriptor … ComplexDescriptor.SimpleDescriptor*** it is possible to declare a tree-path identifying unambiguously each element in a metadata specification.

Table 5 shows the examples of Table 4 with the new pattern. For ISO, the association between classes (e.g. *identificationInfo*) is shown as a

ComplexDescriptor. For FGDC the Metadata root was made explicit. And in the case of DIF the representation structure was collapsed in one line.

Table 5. Identification of metadata descriptors by a tree structure

ISO	MD_Metadata>MD_DataIdentification.citation >CI_Citation.date **MD_Metadata.identificationInfo.MD_DataIdentification. citation.CI_Citation.date**
DCMI	DC.date **DC.date**
FGDC	Identification-Information_Citation_Citation-Information_Publication-Date **Metadata.Identification_Information.Citation. Citation_Information.Publication_Date**
DIF	DIF Data_Set_Citation Dataset_Release_Date **DIF.Data_Set_Citation.Dataset_Release_Date**

An example of the tree components for ISO, FGDC, DC and DIF are presented in the next 4 figures.

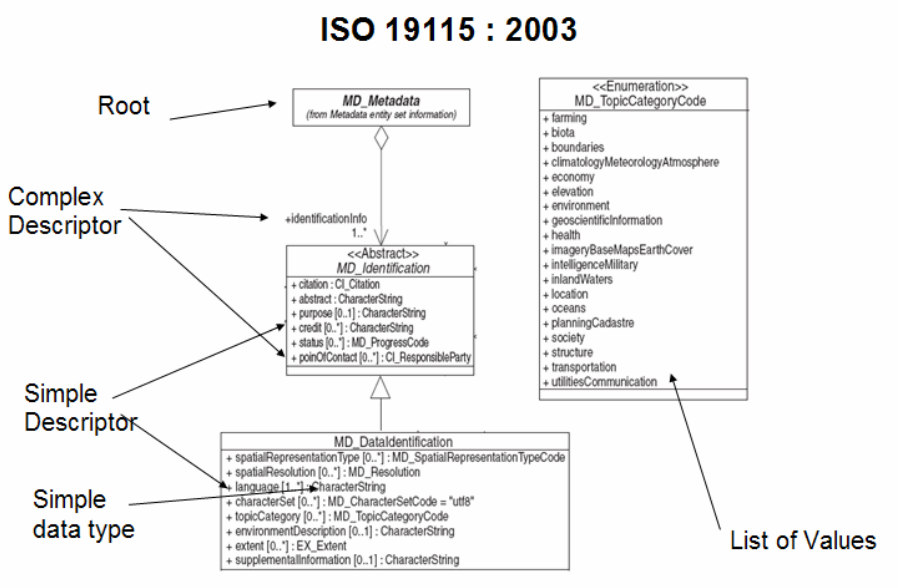


Figure 15. Identification of metadata abstract model components in ISO

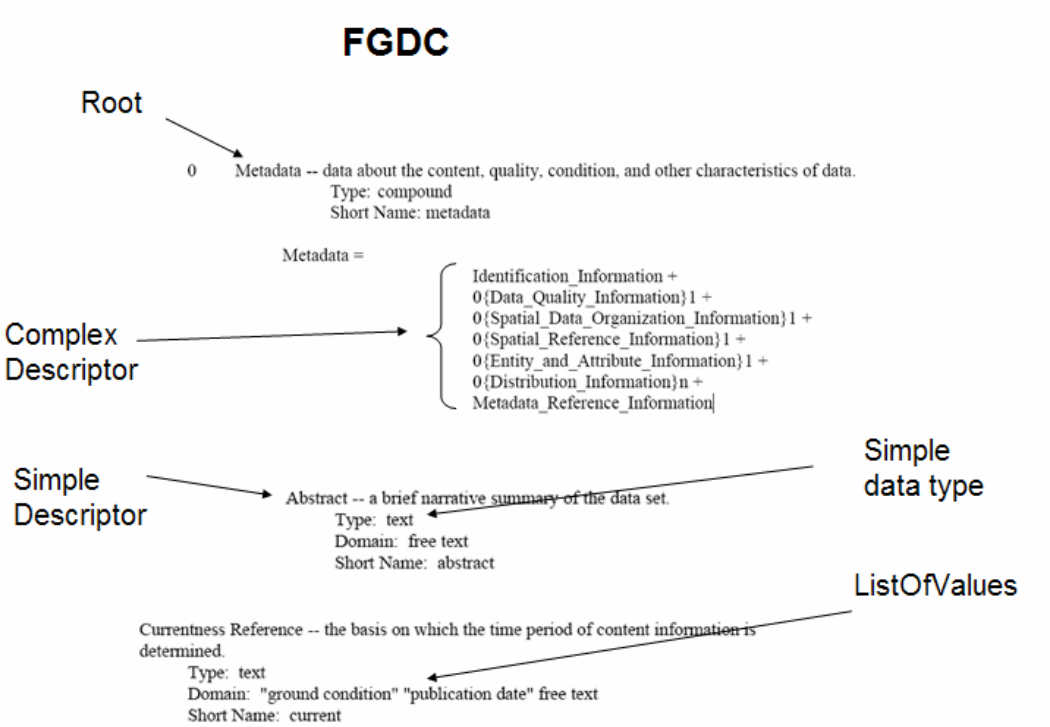


Figure 16. Identification of metadata abstract model components in FGDC

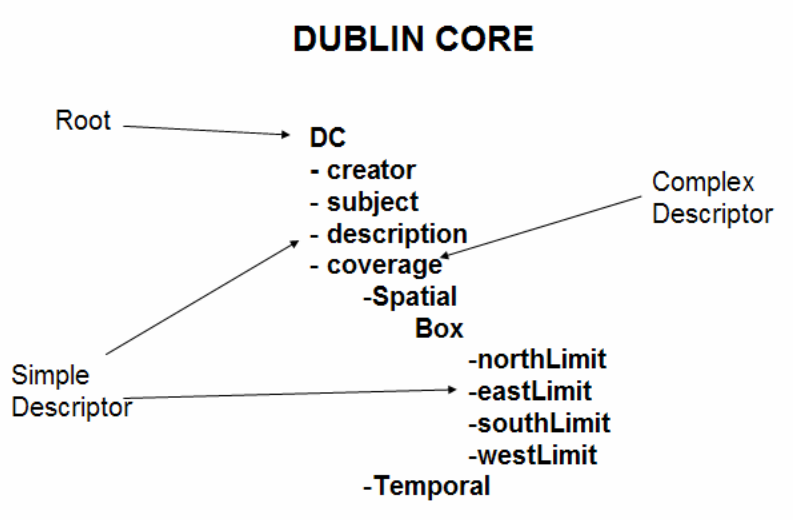


Figure 17. Identification of metadata abstract model components in DC

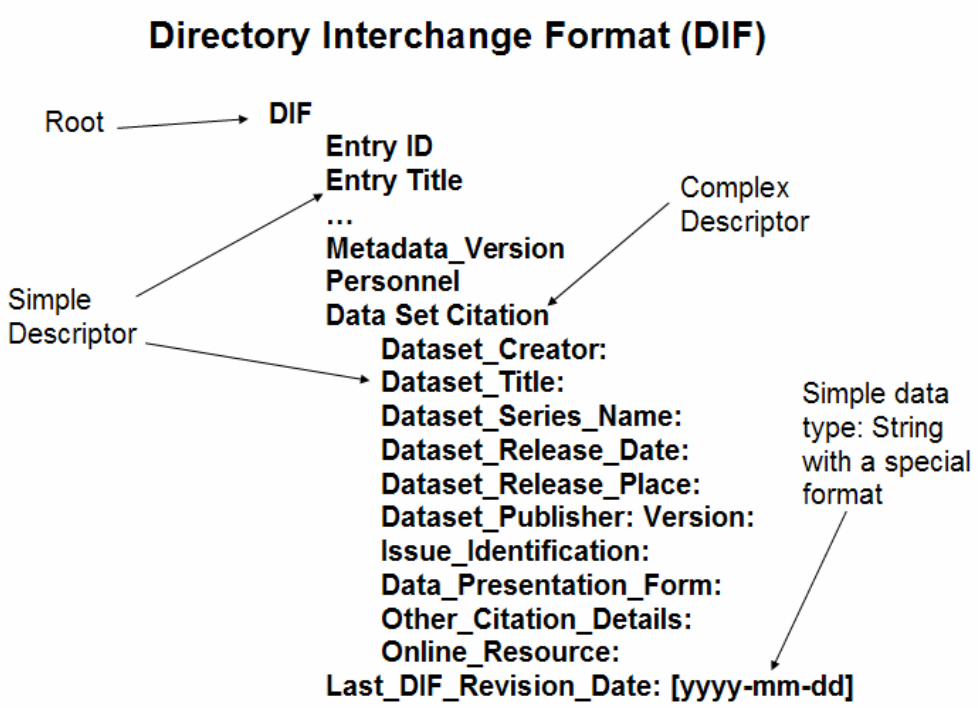


Figure 18. Identification of metadata abstract model components in DIF

A core metadata path can be declared using the tree-path to mark a specific element in a particular hierarchical context. A particular tree-path starts with the root and traverses the tree using a top-bottom approach (parent-child relationship) until it reaches the tail of that particular path. Sometimes it happens that a leaf of a tree is a *ComplexDescriptor*, as it happens when ISO declares the core metadata. It is suggested that the leaf of the tree-path should always be a *SimpleDescriptor*, which in the case of ISO means that it should be an attribute (property).

A MetadataPath has a Path, which is a list of RDF resources which are the URIs of the each descriptor in the TreePath. (See Figure 19).

```
<extender:MetadataPath
rdf:ID="Metadata-identificationInfo-MD_DataIdentification-citation-CI_Citation-date_CI_Citation-CI_Date-date_CI_Date">
  <extender:hasPath rdf:parseType="Collection">
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~how/cuahsi/2004/08/cuahsi#Metadata"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#identificationInfo"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#MD_DataIdentification"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#citation"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#CI_Citation"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#date_CI_Citation"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#CI_Date"/>
    <rdf:Description rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#date_CI_Date"/>
  </extender:hasPath>
</extender:MetadataPath>
```

Figure 19. Metadata path in XML

If metadata specifications are conceptualized in OWL, a more specific tree-path pattern can be depicted. OWL contains Classes and properties. Properties are either objectProperties (i.e. connects two classes) or datatypeProperties (i.e. connects a class with a datatype value). The root of a metadata specification is always a class, and has properties as children. Following the tree-path approach, classes and objectProperties can then be seen as complexDescriptors, while

datatypeProperties can be depicted as simpleDescriptors. The TreePath pattern can be presented in the following way:

Metadata_Root_Class.ObjectProperty.Class.Objectproperty …

Class.datatypeProperty.

A formal grammar definition of a tree is discussed in section 7.3. The next section explains the methodology to create Dynamic Community Profiles using OWL.

6.3 Methodology to create a dynamic metadata community profile:

This chapter presents the methodology to create a Dynamic Community Profile. It will show all the possible extensions one by one, presenting examples using ISO:19115-2003. For ISO the ontology used is the one created by the Drexel Informatics for Civil Engineering (DICE) research group at Drexel University [64].

Typically, all metadata specifications provide guidelines for extending their content and scope as is the case for the ISO-19115 (see Annex C, D and E of the standard). ISO 19115 is presented diagrammatically using the Unified Modeling Language (UML). The ISO 19115 Metadata set is composed of UML packages, each of which is composed of entities (UML classes). An entity contains elements (UML attributes), which identify the discrete units of the metadata. For example, *title*, *alternateTitle* and *date*, are elements of the class *CI_Citation*. To clarify the usage of terms, Table 1 presents the different terminology used by UML, ISO and OWL. UML class is the same as ISO entity and same as an OWL class; and an UML attribute or association, an ISO element and an OWL property refer all to the same concept.

Table 6. Terminology used in OWL, UML, and ISO

UML	Class	Attribute or Association
ISO	Entity	Element
OWL	Class	Property

ISO declares that a community profile should consist of the core metadata set, some optional elements, and newly defined elements. To create a community profile the following extensions are allowed: 1) addition of a new metadata section; 2) creation of a new metadata code list to replace the domain of an existing metadata element that has “free text” listed as its domain value; 3) creation or expansion of a code list; 4) addition of a new metadata element to an already existing entity 5) addition of a new metadata entity 6) imposition of a more stringent obligation on an existing metadata element; 7) imposition of a more restrictive domain on an existing metadata element. The methodology to create the above extensions is discussed in detail in the following sections.

6.3.1 Importing ontologies

The first step for any extension is to import the ontologies that contain the specifications and the vocabularies as shown in Figure 5. The *owl:imports* functions as the extension mechanism and as the link to bridge specifications and vocabularies.

```
<owl:Ontology rdf:about=" ">
   <owl:imports   rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/02/iso-metadata.owl"/>
   <owl:imports   rdf:resource="http://loki.cae.drexel.edu/~how/HydrologicUnits/hu"/>
</owl:Ontology>
```

Figure 5. Ontology extension in XML

The tag *owl:imports* encloses the resource to be imported. Software tools build for the Semantic Web understand the *rdf:resource* tag and will try to load the model located in the URI. Once the model is loaded all the resources contained in the URI are available for extension. The tool, Protégé, does this automatically and displays all the classes and properties of all the imported ontologies, and the JAVA application JENA [61], creates a graph of the imported resource in memory.

OWL syntax conforms to XML namespaces rules, where prefixes and the namespace are declared in the header of the XML file. That is why it is possible to show the abbreviated tag *owl:import*. Also, it is common to create embedded DTDs in the XML file, so instead of presenting the resource: *http://loki.cae.drexel.edu/~wbs/ontology/2004/02/iso-metadata.owl#CI_Citation*, we should be able to present the resource as *&iso;CI_Citation* when it is used as an attribute value. We will follow this abbreviated convention in the subsequent examples to improve readability.

From now on, the term ***element*** will be used to denote a property in a metadata specification formalized in an ontology. An element can be an objectProperty as well as a datatypeProperty.

6.3.2 Setting an element as core

A community determines the elements that constitute its profile by marking elements of another specification as core. A core element can be mandatory or optional and it should always be available when a metadata instance is going to be created or when there is a need of semantic validation or when a metadata repository is going to be queried.

There are three possible ways to specify (or mark) core elements: 1) setting all the other non-core elements with cardinality equal to zero, 2) creating a flag on the elements that must be core, and 3) using a tree-path approach.

In the first case an algorithm is used to discard elements with a zero cardinality. The process of creating a core profile using a conceptualization tool like OWL, that allows to create subclasses and restrict inherited properties, is as follows:

1. Create a subclass for all the classes in the original specification.
2. Create a restriction cardinality equal to zero for all the elements.
3. When an element is marked as core, the previously created restriction cardinality is deleted.
4. When an element is marked as not core, the class of that element is sub classed and a property restriction is created with cardinality equal to zero.

In the second case, creating a flag to tell a system that an element is core can be done by creating an annotation property named “isCore”. If its value is “true”, then it is core, otherwise if its value is “false” then the property is not core.

The previous two approaches (restriction cardinalities and flags), to set up core elements will work only if elements are not reused inside the metadata specification. For example, classes that are range of more than one property (hereafter called multi-range-class) are consider reused elements. Figure 20, shows a multi-range-class in ISO-19115, MD_identifier, which is a range of two properties EX_GeographicDescription.geographicIdentifier, and CI_Citation. identifier.

To illustrate the problem let us assume that we want to set *authority* in MD_Identifier as core, but only when it is used in the context of EX_GeographicDescription.geographicIndetifier. Using the cardinality restriction or flag, MD_Identifier.authority can be declared as a core element. However, both EX_GeographicDescription.geographicIndetifier and CI_Citation.identifier are affected by the declaration, and as a result MD_Identifier.authority is true in all the context that it is used. All the ISO classes that were mapped from a UML class, with a stereotype datatype, to an OWL class are exposed to this problem.

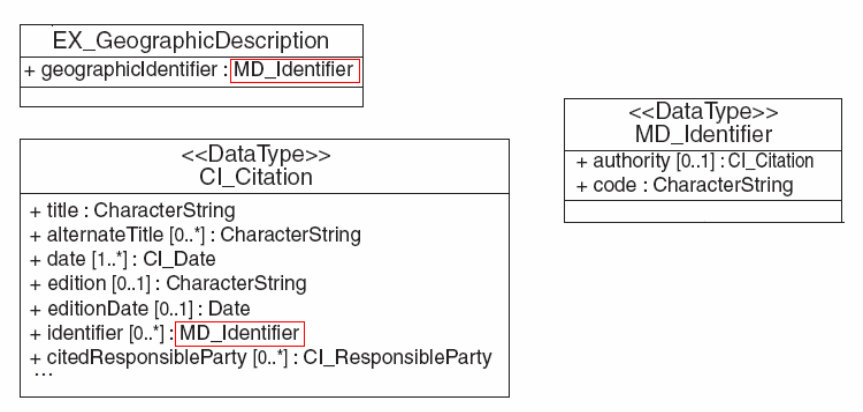


Figure 20. Multi-range-class

Because of the problem with multi-range-classes, a new methodology is created based on the tree-path model previously explained. The proposed model guarantees that the element being referred to is exactly an element in a particular context, and not the element in all the contexts of the specification. The metadata-tree-path for MD_Identifier.authority in the context of EX_GeographicDescription is as follows:

MD_Metadata . identificationInfo . MD_DataIdentification . Extent . EX_Extent. geographicElement . EX_GeographicDescription . geographicIdentifier . MD_Identifier . authority

6.3.3 Unsetting an element as core

An element in the specification is not core by default. However if a previous element was set as core, it should have a metadata-tree-path declared as core. To unset a path previously set as core, the path can be declared un-core, or simply deleted.

6.3.4 Setting as mandatory

Setting an element as mandatory means that when its parent is used, this element should be annotated. When a specification is presented in a conceptual way, properties have associated cardinalities related to a particular class. For example, *Citation* is a class that has a property *title*. *Title* has cardinality one, which means that *Citation* should have only one title.

Setting an element as mandatory in OWL is done by setting the cardinality to one or more, or setting the minimum cardinality to one or more. In OWL cardinalities can be created (or *overwritten*) using restrictions over properties. A

restriction in OWL is a super-class of a class that holds the property. To solve potential conflicts that appear when a subclass declares a different cardinality than its parent, the subclass declaration always prevails among the super-class declaration. More about restrictions is discussed in 6.3.9.

6.3.5 Adding a new metadata section or metadata entity

A metadata section is a set of classes that are related to each other. In OWL, however, there is no direct means to create a package or section. One way to separate the original classes from a new set of classes is to create a new ontology that resembles a package. In OWL, classes are created using the *owl:Class* tag. The *rdf:ID* is the identifier of the resource as shown in Figure 21.

```
<owl:Class rdf:ID="gic;MD_Keywords_ext">
    <rdfs:subClassOf rdf:resource="&iso;MD_Keywords"/>
</owl:Class>
```

Figure 21. Creation of a class and a sub-class in OWL

The newly created class or classes need to be linked with the original model. This is done by creating a new property and declaring the domain to be a class in the original model or by creating a new parent-child relationship that links a class from the original specification to the newly created class in the profile. Creation of a new property is explained in section 6.3.8 below and creating a new parent-child relationship is done by specifying that the new class is a subclass of the original one, as shown in the right panel in Figure 21. In the above example, the class *MD_Keywords_ext* is a new class that is a subclass of M*D_Keywords*, which is an original ISO class.

6.3.6 Creating a new metadata code list

Code-lists and enumerations in ISO are defined as datatype classes whose instances form a list of named literals that contain a set of values. In OWL this can be represented as a class whose instances are the list of possible values. Figure 22 shows an excerpt of a list of gauge stations for the Neuse River Basin, encoded in XML, where a new class *Station* and instances of those classes are created with their respective ids and labels in English (=”en”).

```
<owl:Class rdf:ID="Station">
  <rdfs:label xml:lang="en">Station</rdfs:label>
</owl:Class>
<Station rdf:ID="_02085500">
   <rdfs:label xml:lang="en">FLAT RIVER AT BAHAMA, NC</rdfs:label>
</Station>
<Station rdf:ID="_02087570">
  <rdfs:label xml:lang="en">NEUSE RIVER AT SMITHFIELD, NC</rdfs:label>
</Station>
```

Figure 22. Neuse-Station ontology

The *station* class in the previous ontology excerpt can be used as the range of a property in another ontology. In Figure 23, the range of the property *site* refers to all the stations located in the Neuse-station ontology in Figure 22 or simply the instances of *&neuse;Station*.

```
<owl:ObjectProperty rdf:ID="site">
  <rdfs:domain rdf:resource="&iso;MD_DataIdentification"/>
  <rdfs:range rdf:resource="&neuse;Station "/>
</owl:ObjectProperty>
```

Figure 23. Usage of a Code-list as a range of a property in OWL

6.3.7 Expanding a code list

Code-lists, as previously mentioned, are formalized in OWL as a class. To add a new member of the class one needs to create a new instance of that class. An ISO code-list will look similar to Figure 24.

```
<owl:Class rdf:ID="TopicCatCd"/>
  <TopicCatCd rdf:ID="_005">
    <rdfs:label xml:lang="en">economy</rdfs:label>
  </TopicCatCd>
  <TopicCatCd rdf:ID="_012">
    <rdfs:label xml:lang="en">inlandWaters</rdfs:label>
  </TopicCatCd>
  ....
```

Figure 24. ISO Code-list

After importing this ontology an instance can be created for one of the imported classes. Figure 25 shows a new instance of the class TopicCatCd with ID = “_020”.

```
<TopicCatCd rdf:ID="_020">
  <rdfs:label xml:lang="en">groundwater</rdfs:label>
</TopicCatCd>
```

Figure 25. Extending an ISO's code-list.

6.3.8 Adding a new metadata element to an existing class

A metadata element is equivalent to a property in OWL. In OWL there are two types of properties: datatype properties (*owl:datatypeProperty*) and object properties (*owl:ObjectProperty*). The datatype property has as its range an XML datatype (e.g. string, integer, date and others given by the XML schema

specification), while an object property has as its range an *owl:Class*. Figure 26 shows the creation of a new property, named *site*, whose domain is the ISO *MD_DataIdentification* class, and whose range are the instances of the class Station as previously defined in the Neuse-Station ontology.

```
<owl:ObjectProperty rdf:ID="site">
  <rdfs:domain rdf:resource="&iso;MD_DataIdentification"/>
  <rdfs:range rdf:resource="&neuse;Station "/>
</owl:ObjectProperty>
```

Figure 26. Creation of an object property in OWL

If we would like to create a property whose range is a simple data type, like an integer, then a owl:*datatypeProperty* should be declared as shown in Figure 27:

```
<owl:DataProperty rdf:ID="siteNumber">
  <rdfs:domain rdf:resource="&iso;MD_DataIdentification"/>
  <rdfs:range rdf:resource="http://Web.w3.org/2001/XMLSchema#int"/>
</owl:DataProperty>
```

Figure 27. Creation of a datatype property in OWL

6.3.9 Imposing a more stringent obligation on an element.

Restricting a metadata element can be interpreted as imposing restrictions on a property in OWL. The restrictions that are available in OWL are: *hasValue*, *allValueFrom*, *someValuesFrom*, *minCardinality*, *maxCardinality*, and *cardinality*. All of these restrictions can be applied to extend properties, shaping the extended ontology to fit specific needs of GICs.

A property has the following characteristics: cardinality, type, and range. In order to change the characteristics of a property the following must be done: first, create a subclass of the class that holds the property to be extended, and second, create a local restriction to the extended property in the subclass.

Imposing a more stringent obligation on an existing metadata element requires to “change” the cardinality of an element. In OWL this is done by creating a local restriction on a property. Since it is a more stringent obligation, this means that the cardinality before was zero and now must be one. This can be done by stating that *owl:minCardinality* or *owl:cardinality* is equal to one. Figure 28 shows the XML expression for the ISO element *datasetURI*.

```
<owl:DatatypeProperty rdf:ID="&iso;dataSetURI">
  <rdfs:label xml:lang="en">dataSetURI</rdfs:label>
  <rdfs:domain rdf:resource="&iso;MD_Metadata"/>
  <rdfs:range rdf:resource="http://Web.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Figure 28. ISO element datasetURI

If this property is to become mandatory, the following two steps must be followed: first, create a new subclass of *&iso;MD_Metadata*: *&ext;MD_Metadata_ext,* and second, a restriction; *owl:minCardinality* on the property *&iso;dataSetURI*. An example is shown in Figure 29.

```
<owl:Class rdf:ID="&ext;MD_Metadata_ext">
    <rdfs:subClassOf rdf:resource = "&iso;MD_Metadata">
        <rdfs:subClassOf>
            <owl:Restriction>
             <owl:minCardinality rdf:datatype="&xsd;int"> 1 </owl:minCardinality>
             <owl:onProperty>
                  <owl:DatatypeProperty rdf:about="&iso;dataSetURI"/>
             </owl:onProperty>
            </owl:Restriction>
 </rdfs:subClassOf>
 ...
</owl:Class>
```

Figure 29. Cardinality restriction in OWL

6.3.10 Imposing a more restrictive domain on an element.

Imposing a more restrictive domain on an existing metadata element is interpreted in OWL as changing the range of a property. It is similar to the previous case because it implies creation of a local restriction on a property.

Figure 30 shows a graphical view of changing the domain of *iso:keyword* on the class *MD_Keywords* and Figure 31 shows the proposed extension in XML. The extension implies that the range (or domain in ISO) of the property *iso:keyword* is no longer *CharacterString* but now contains all values from *gcmd:Surface_Water.* In OWL, such restrictions are done indirectly by stating that the class that is restricting the property is a subclass of a class called *owl:Restriction.* The *owl:Restriction* contains the property that is being restricted, *iso:keyword,* and the type of restriction, *owl:AllValuesFrom*.

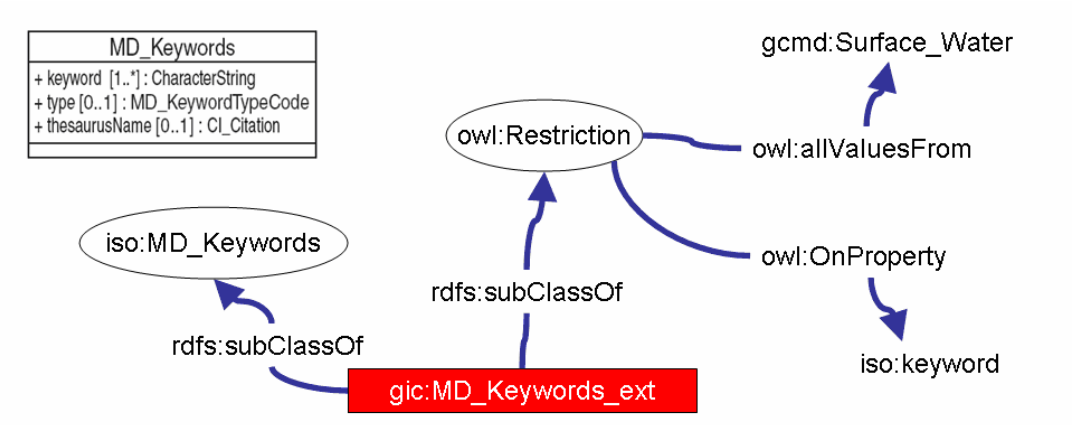


Figure 30. Restriction iso:keyword

```
<owl:Class rdf:ID="gic:MD_Keywords_ext">
    <rdfs:subClassOf rdf:resource="&iso;MD_Keywords"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="&iso;keyword"/>
            <owl:allValuesFrom rdf:resource="&gcmd;Surface_Water"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Figure 31. Extension of iso:keywords in XML

The *owl:ValuesFrom* tag is a built-in OWL property that links the restriction to a class description or a data range. A class description is a defined class, whose instances are all the values that the restrictions refer to. In Figure 31 this class is *gcmd:Surface_Water,* which contains instances like: *discharge* and *stage height*. It is important to note that the logical reading of the created statement is "all individuals that have values for the property *iso:keyword* of type *gcmd:Surface_Water* are of type gic:*MD_Keywords_ext*".

6.3.11 Extension problems

6.3.11.1 Restricting datatype properties

OWL can be used in three different versions: OWL-Light, OWL-DL and OWL-Full as mention before. While the OWL-Full version is the most expressive, the other two guarantee computational completeness and decidability. The metadata specification as well as the controlled vocabularies and the extensions where created in OWL-DL only; however, there are some expressions where OWL-Full is needed, as for the case presented previously in Figure 31. The figure shows a restriction that uses *allvaluesFrom*. If the property to which the restriction is applied is a datatype property and the restriction on that property is not a *datatype* class, or *rdfs:literal*, or a *oneOf*, it becomes an OWL-Full expression. Since *gcmd:Surface_Water*, in Figure 31 is an owl:Class and not a datatype class, or a rdfs:literal or a oneOf, the statement is in OWL-Full.

There are two possible solutions to this problem: 1) Change the datatype property to be an object property in the original metadata specification. This is not feasible, since the metadata specification should be an independent Web-accessible resource, published by an entity (e.g. ISO) that is different from the community creating the extension. For obvious reasons the community can not be permitted to change the original metadata specification directly, because this specification will be used by other communities. 2) Treat the datatype property as an object property in the extension. In this case the extension will be in OWL-Full. Because of the concern with regard to the computational completeness and decidability when using an ontology in OWL-Full, we tested our extension using *Pangloss*, which is the suite of tools created to test the methodologies proposed in

this thesis. It was found that it was possible to create such expressions without encountering any problems.

An instance can be presented in two ways as an *objectProperty* or as a *datatypeProperty* as presented in Figure 17. Both have the same meaning in an RDF graph; however, a computer application that reads these instances should be able to accommodate instances in which some literal values might be a URI (e.g. “http://foo#b1), as shown in the right panel of Figure 17.

As object property:

```
<A rdf:ID="individual_a">
   <hasB rdf:resource="http://foo#b1"/>
</A>
```

As datatype property:

```
<A rdf:ID="individual_a">
   <hasB>http://foo#b1<hasB>
</A>
```

Figure 32. Value of a property as object Property and as datatype Property

6.3.11.2 Restricting inherited properties on classes that are the range of more than one property

On occasions the need arises to restrict a property in multi-range-class (See Figure 20). If a multi-range-class is restricted, every time the class is being used (e.g. range of a property) the restriction will apply. If the *MD_Identifier* of *geographicIdentifier* is restricted to have only one possible value for its authority property (e.g. the citation referring to GETTY, which provides a thesaurus for geographic names), then *identifier* in *CI_Citation* will also be subject to the same restriction. The workaround of this problem is to also create an extension on the domain of the property that uses a multi-range class and apply the restriction to that class. Figure 33 depicts the schematic where the class

ext:EX_GeographicDescription_ext, which is a subclass of *iso:EX_GeographicDescription,* is subjected to a restriction with allValuesFrom *ext:MD_Identifier_ext*. Using this procedure a computer program can be coded to prefer the extended classes, from the original ones, and validate the instances, display a user interface or query metadata instances that conform to a particular GIC.

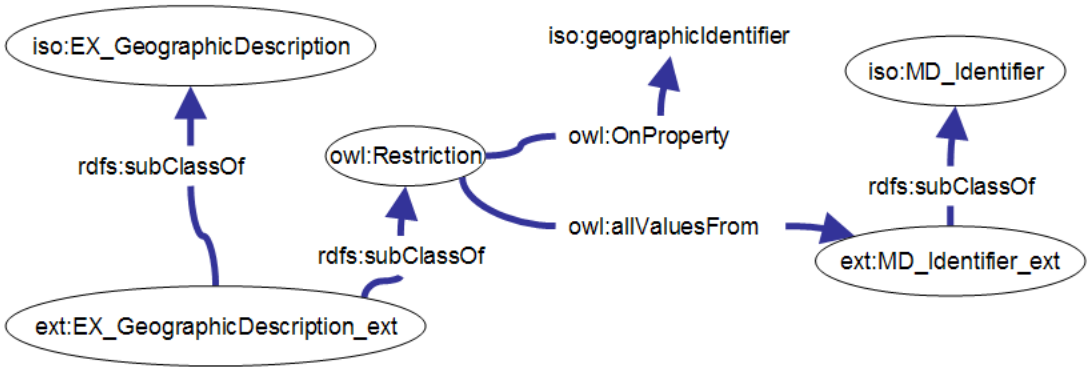


Figure 33. Extending a property of a multi-range-class

6.4 Related approaches to Dynamic Community Profiles

Dynamic Community Profiles are an extended notion of community profiles [32, 65] and application profiles [4, 53]. Community profiles, provide an extension of metadata standards to fit specific community needs, but they tend to be monolithic (i.e. not allowing combination with other specifications).

Application profiles [53] rely on the Resource Description Framework to reuse elements of several metadata specifications, which are identified with unique namespaces. This approach allows to import other schemas and use elements defined in the imported schemas. Extensions that can be performed under RDF

schemas include: defining of a new element and create a list of controlled vocabulary to use as annotation of the description. A new element is defined by creating a schema in a new namespace, and declaring RDF properties in that schema. A controlled vocabulary for an element is defined by creating a new class and creating instances of that class. To link the controlled vocabulary to a property, the class is declared to be a range of that property.

Activities that use RDF schemas, normally use DCMI plus other domain schemas that specify a set of properties and controlled vocabularies. The Development of a European Service for Information on Research and Education (DESIRE) allows registering and mapping of RDF schemas (and namespaces) related to library and education information [126], and currently it has approximately 20 application profiles. Other approaches that use an extension of DCMI, but that are not related to library and education are: HealthCybermap [66] and RDFPIC [70]. The former is related to health metadata while the latter is related to photographs.

RDF Schemas allow creation of classes and properties but are not sufficiently flexible to allow expression of extensions given in geographic metadata specifications such as ISO-19115:2003. Redefining the range of a previously created property or changing the cardinality or obligation it is not possible when using the vocabulary provided in the RDF schemas. One approach is to create another schema defining new properties that are used to specify the profiles and that allow overwriting properties. Baker et. al. [4] propose an extension mechanism to redefine elements, defining a new property called “uses”. This property is used to declare the profiles. Also, it is permitted to overwrite the domain, changing the label

and comments to attain a more comprehensive profile. Figure 34 shows a special property *sf:uses*, which marks selected elements to be used in a profile. The figure also shows a redefinition of the label and the comment of *dcq:temporal* element.

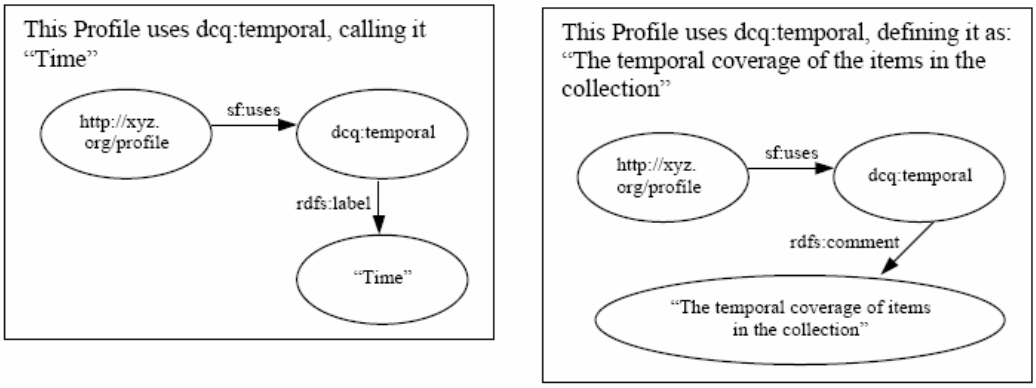


Figure 34. Extracted Figure 3 and 4 from [4]

The overwriting mechanism in this thesis is different from that of Baker et. al., in that instead of declaring new semantics to overwrite characteristics of elements, the restriction capabilities of OWL are used. Also, to specify core elements, instead of the "uses" property of Baker, a metadata-tree-path is used, as explained in Chapter 6.2. The metadata-tree-path model allows a more general approach for refining metadata elements and provides a solution for the multi range classes problem which is not possible with Baker's strategy.

This chapter presented the methodology to create Dynamic Community Profiles. First it discussed an abstract model, where metadata specifications can be seen as tree-paths. Then it is explained the selection of elements from other specifications by setting then them as core; addition of new elements; creation and expansion of code lists; and imposition of more stringent obligations and more

restrictive domain on elements. It also explained extension problems and related approaches using the Resource Description Framework.

CHAPTER 7: MAPPING METADATA SPECIFICATIONS

7.1 Motivation and background

Agreements in metadata specifications are necessary to achieve semantic interoperability among communities. However, more than one metadata specification exists and it is impossible to achieve a general consensus about the use of one and only one specification. To minimize the time spend in maintaining content metadata and to maximize the use among a broad range of users, metadata in one specification must be available in other related specifications [87, 115]. Therefore, metadata crosswalks are necessary to facilitate interoperability among communities which uses heterogeneous and related specifications.

A crosswalk is an explicit mapping of one metadata specification to another. Creating a crosswalks is a difficult and error-prone task, which requires in-depth knowledge and experience of the metadata specifications being mapped [115]. Developing a crosswalk between two metadata specifications requires the following steps [87]:

1. Harmonization: The two metadata specifications are expressed in the same format or model.
2. Semantic mapping: Elements from a metadata specification are explicitly mapped to one or more elements of another metadata specification.
3. Additional rules: Necessary to solve complex mapping such as hierarchy level or one-to-many mappings.
4. Mapping implementations: The process of mapping a metadata instance from one format to another following the semantic mappings and the rules.

Semantic mappings are usually presented informally in HTML files (e.g. GCMD), specifications annexes (e.g. ANZLIC) and in tables (e.g. DGIWG, [87]). Informal mappings include mappings among labels or ids. For example, the Global Change Master Directory (GCMD), presents conversions from the Directory Interchange Format (DIF) in http://gcmd.gsfc.nasa.gov/Aboutus/standards/ to ISO, FGDC, GeoConnections, Z39.50, Catalogue Interoperability Protocol (CIP), National Biological Information Infrastructure (NBII), Dublin Core Metadata Initiative (DCMI) and the Australia New Zealand Land Information Council (ANZLIC). However, the trend is that geospatial metadata specifications should have interfaces and crosswalks to ISO 19115:200, which can be discerned from metadata and interoperability studies (e.g. [29]), and from public communications or annexes of other publisher of standards (e.g. FGDC, ANZLIC) .

Some efforts to coordinate different metadata specifications are provided by groups such as Digital Geographic Information Working Group [22], which offers a crosswalk between the ISO and the FGDC standards. These crosswalks are accepted and recommended by FGDC in their FGDC/ISO Standard Harmonization Web site [33]. A compilation of crosswalks is presented by the Metadata Architecture and Application Team (MAAT) in http://www.sinica.edu.tw/~metadata/tool/mapping-foreign.html. MAAT is a group affiliated with the National Digital Archives Program in Taiwan. Their purpose is to support metadata implementation projects and by developing a Metadata Framework Model. (For more references about the topic see [87]).

Formal crosswalks implementations, usually are hard coded in software programs (e.g. [119]), or expressed in XSLT (e.g. [23, 87]). XSLT [16] is based on

the eXtensible Style language, which is used to create styles for XML documents and allows formatting from one format to another. An XSLT crosswalk form DCMI to ISO can be found in [87], and a transformation from the ADN to DCMI and vice versa can be found in [23]. The later includes both, a transformation of the label of the elements as well as transformations of the controlled vocabulary used. The transformation is possible from an ADN instance in an XML document to an instance in DC in an XML document For example:

<general><title>Hello World</title></general>

GOES TO

<dc.title>Hello World</dc.title>

AND

<educational>

<resourceType>

DLESE:Text:Reference

</resourceType>

</educational>

GOES TO

<dc.type>Text</dc.type>

The XSLT is the most sophisticated tool to transform one XML file format to another; however, the mappings of the semantics is not clearly expressed, since the semantics and the syntax transformation rules are both declared at the same time. This is the reason why, semantic mappings are written in static tables (e.g.

[22, 87]) which only serve the purpose of guidance when an individual is creating the XSLT.

This chapter discusses an approach to formally express semantic mappings following a model for metadata crosswalks, based on tree-paths discussed earlier. The model was integrated in a tool to provide a view of the semantic mappings dynamically. Also a 1D transformation was performed from FGDC to ISO. The 1D transformations are those where the intermediate nodes are not repeated.

7.2 Process overview

1. Harmonization: The source and target metadata specifications are both expresses in a conceptual form in OWL. (See Chapter 8). And the model, where the mappings take place, is a tree with special characteristics expressed in the next section.
2. The semantic mappings are express as a *MetadataMapping*, where there is a *toPath* and a *fromPath*. These to Paths are *MultiMetadataPaths*, composed of one or more paths. Each path is composed of a value and a list of resources. Mappings flags and code handler, can be used to perform rules in complex mappings. For example, one-to-many mappings where the syntax should be known and should have a special treatment. All the mappings are stored in a *MetadataMappingModel*, as shown in Figure 35.
3. Rules: The rules should be specified in the model, to have them all expressed in machine-readable format.
4. Transformation process algorithm is expressed in the next section.

The next section presents a formal definition of a tree, the components of the metadata mapping model, and the transformation procedure.

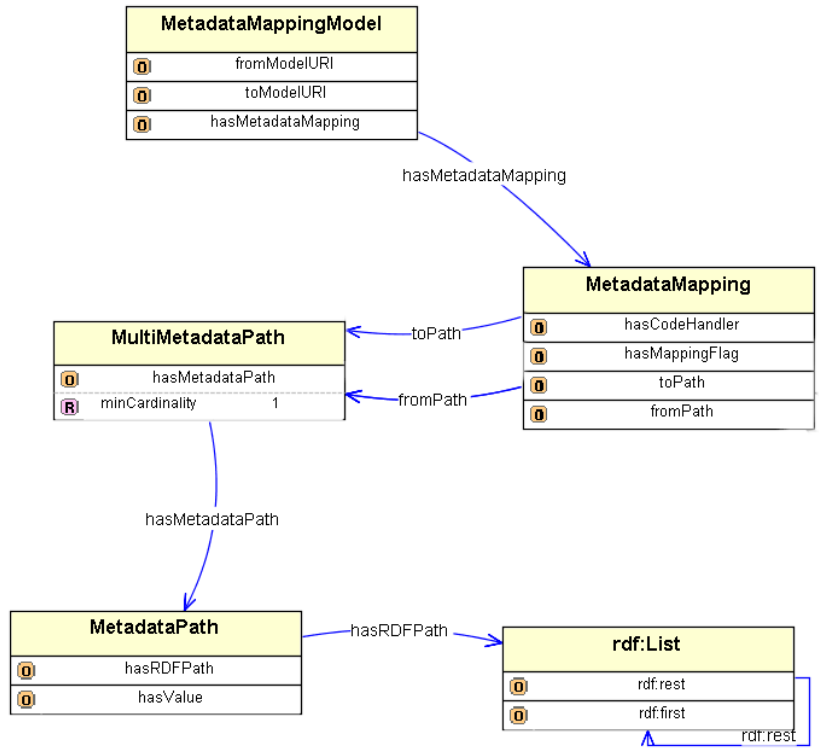


Figure 35. Metadata Mapping Model

7.3 Metadata-Tree-Path

Metadata specification is a rooted tree. A rooted tree has the following properties:

- It is composed of vertices (nodes) and graphs (arcs). A node can be either a classes or a property. And an edge represents either the class-property or property-class relationship.
- It contains only one root, which is a node and a class.
- Two vertices are connected by at most one path.

- It is undirected, meaning that no direction is specified.
- Has no simple cycles. Two nodes that apparently are the same, For example, having the same URI, can be found in one branch. But they have different node number, so they are different, not allowing cycles.
- If any edge is removed, it is not connected anymore.
- The number of edges is equal to number of nodes – 1.

A metadata tree can be seen as a set of paths. A path is defined as: $p = \{n_1, n_2, ...n_m\}$, where $n_1 = root$ and $n_m = leaf$, and *m* is the number of nodes in the path. A path can have a value, which is the value of the leaf. A path with value is defined as:

$$p = \langle \{root, n_2, ...n_m\}, val \rangle .$$

- A multipath ($p*$) is a set of one or more paths, $p* = \{p_1, p_2, ...p_n\}$, where *n* is the number of paths contained in the multipath.
- A metadata specification is defined as $S = \{p^{S*}{}_1, p^{S*}{}_2, ... p^{S*}{}_n\}$, where *n* is the number of multipaths contained in the metadata specification.
- A metadata instance is defined as $I = \{p^{I*}{}_1, p^{I*}{}_2, ... p^{I*}{}_n\}$, where *n* is the number of multipaths contained in the metadata instance.
- I conforms to S if for all $p^{I*}{}_1, p^{I*}{}_2, ... p^{I*}{}_n$, define in I there is a similar path in given specification S_i. This will be denoted as I^{S_i} .
- An instance path that conforms to one path in *S* is denoted as $P^{I \subset S_i}$

- A mapping definition $M_{(S_i \to S_j)_k}$ from any specification S_i to any specification S_j is define as a pair of multipaths. The first one, $p_k^{S_i*}$ depicting the ***source*** and the right one, $p_k^{S_j*}$, denoting the ***target***: $M_{(S_i \to S_j)_k} = \left\langle p_k^{S_i*}, p_k^{S_j*} \right\rangle$, where *k* is greater than zero and less than or equal to the total number of mappings. The set of all define mappings is define as $M_{(S_i \to S_j)}$.
- A mapping function is defined as: $MAP(P_i^{I \subset S_i*}, M_{(S_i \to S_j)}) = P_i^{I \subset Sj*}$. The mapping is performed using a specified set of mappings, $M_{(S_i \to S_j)}$, and takes a multipath instance, $P_i^{I \subset S_i*}$, which conforms to S_i and translates it to a multipath instance $P_i^{I \subset Sj*}$ which conforms of a target specification, S_j. Table 7 shows different possible mappings.

Table 7. Possible Mappings

FROM \ TO	**ONE**	**ONE AND ONE WITH VALUE**	**MANY**	**MANY AND ONE WITH VALUE**
ONE			**SYNTAX**	**SYNTAX**
ONE AND ONE WITH VALUE			**SYNTAX**	**SYNTAX**
MANY	**SYNTAX**	**SYNTAX**		
MANY AND ONE WITH VALUE	**SYNTAX**	**SYNTAX**		

- A ONE_TO_ONE mapping is when the source and target multipaths has one path with no value defined. (e.g. *fgdc:Metadata_date* maps to *iso;datestamp*)
- A ONE_TO_ONE_AND ONE_WITH_VALUE, occurs when the source has one path with no defined value and the target has one path without a value and a path with a defined value. (See Figure 36)

Figure 36. A ONE_TO_ONE_AND ONE_WITH_VALUE mapping

The areas not shaded in Table 7 are not well-formed mappings, and can be defined as two or more well formed mappings (shaded area). For example, many to many mappings can be resolved with two or more ONE_TO_ONE mappings. A mapping with the SYNTAX word, means that further knowledge of the source and

target syntax is necessary. For example, knowing if the values must be separated with a comma or a tab character.

A transformation from one tree-path to another can be performed by an algorithm that iterates over the mappings, gets the value from the source path and copy this value in the target path. To explain the algorithm the following is assumed:

- A source metadata specification : S_i .
- A target metadata specification: S_j .
- A metadata instance: I^{S_i} , which conforms to S_i , the instance that contains the source.
- A metadata instance I^{S_j} , which conforms to S_j , the instance where the values are going to be mapped to.
- Set of mappings: $M_{(S_i \to S_j)}$ from Specification S_i to S_j . Which has a total of mappings = *totalMappings.*

The main method to perform the mapping is as follows:

int k = 1;
While (k <= totalMappings) {
// get he kth mapping from mappings
$M_{(S_i \to S_j)_k}$ *= getMapping (* $M_{(S_i \to S_j)}$ *, k)*
//Get the source multipaths
$p_k^{S_i^*}$ *= getSourcePaths (* $M_{(S_i \to S_j)_k}$ *)*
//get similar multipaths from an instance conforming to Si
$P_k^{I \subset S_i^*}$ *= getSimilarPaths (* I^{S_i} *,* $p_k^{S_i^*}$ *),*
totalPaths = getTotalPaths ($P_k^{I \subset S_i^*}$ *)*
int m = 1;
while (m <= totalPaths) {
// get path from source instance in position m

$P_m^{I \subset S_i}$ = *getSourcePath(* $P_k^{I \subset S_i^*}$, *m)*

// get the value in the source path

$v_1 = getValue(p_1^{I \subset S_i})$

If (is_ONE_TO_ONE($M_{(S_i \to S_j)_k}$ *)) {*

// get the first path in the target

$p_1^{S_j} = getFirstPathTarget(p_k^{S_i^*})$

// assign value to a path x

$p_x^{I \subset S_j} = assignValue(p_1^{S_j}, v_1)$

}

else

If *(is*_A ONE_TO_ONE_AND ONE_WITH_VALUE*(* $M_{(S_i \to S_j)_k}$ *)){*

// get the path target with no value

$p_1^{S_j} = getPathTargetNoValue(p_k^{S_i^*})$

// assign value to a path x

$p_x^{I \subset S_j} = assignValue(p_1^{S_j}, v_1)$

// add the path x to the target instance

$p_{temp}^{I \subset S_j}$ = *add (* I^{S_j} , $p_x^{I \subset S_j}$ *)*

// get the path with defined value and added

$p_2^{S_j} = getPathTargetWithValue(p_k^{S_i^*})$

$p_{temp}^{I \subset S_j}$ = *add (* I^{S_j} , $p_2^{S_j}$ *)*

}

}

m++;

}

k++

The semantic mapping can be stored in XML-RDF, following the model presented in Figure 35. A result of a mapping from an instance in FGDC to ISO is shown in Figure 37.

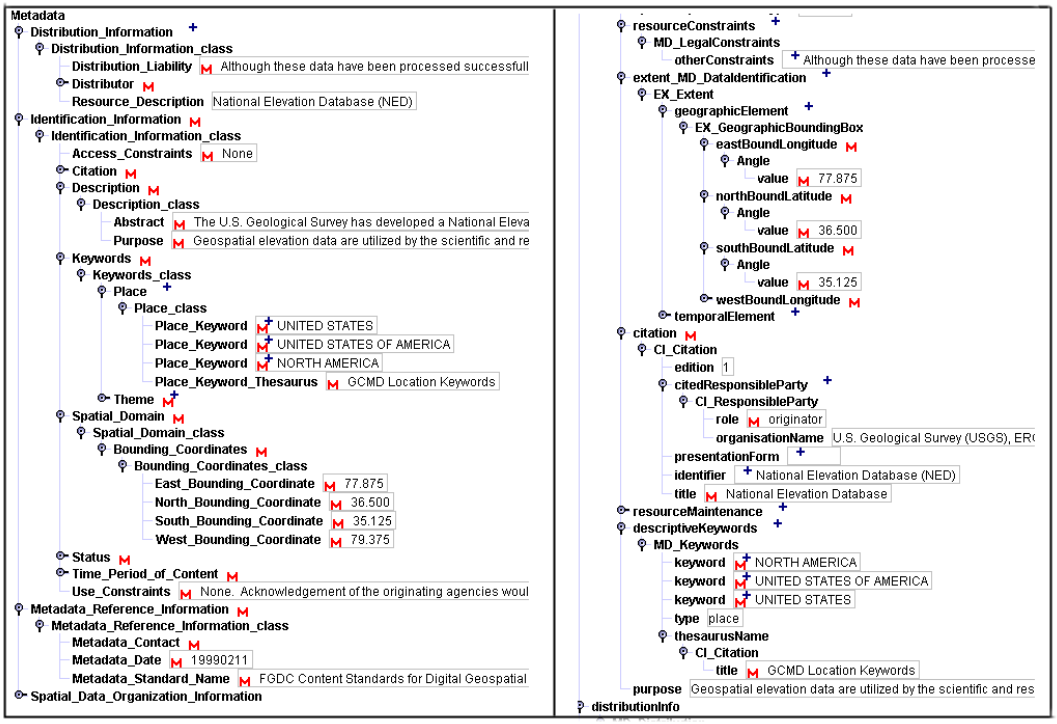


Figure 37. Mapping example from FGDC to ISO

The approach presented in this chapter works for 1D mappings. 1D mappings are those where the intermediate nodes are not repeated. If they are repeated, they need a further check in the algorithm. The processed source and target paths need to be stored in a temporary map, and every time a repeated path is going to be processed it will check the temporary map to find the exact node of replication on the target instance.

This chapter presented the importance of metadata mappings, the process of mapping metadata specifications and a formalization model based on tree-paths. Also, it was discussed a categorization of all the possible mappings and algorithms to perform the mappings based on the formalized model.

CHAPTER 8: CONCEPTUALIZATION OF METADATA SPECIFICATIONS IN OWL

At the time of writing this thesis there was no endorsement by ISO, or any other publisher of standards, of a metadata specification formalized in OWL. However, communications with members of the Technical Committee 211, TC211, of ISO suggest that efforts are underway to consider an endorsement of the ISO 19115:2003 in OWL in the near future. While this may be in part due to the fact that OWL is in its infancy (newly recommended by W3C, 10 February 2004) a mapping of several metadata specifications created by Drexel University are available at: http://loki.cae.drexel.edu/~wbs/ontology/list.htm. This chapter discusses some of the strategies taken to conceptualize metadata specifications in OWL, starting from two possible scenarios: 1) a specification conceptualized in UML and 2) a specification not conceptualized. For the former a conversion from UML to OWL of ISO 19115-2003 is presented and for the later a conversion from FGDC in DTD format to OWL is discussed.

8.1 ISO in OWL

Conversion of ISO metadata specification from UML to OWL is discussed in [64]. A summary of the mappings are presented Table 8 and

Table 9. The most important conversions are: A UML class is mapped to an owl:class; a UML association is mapped to an owl:ObjectProperty; a UML attribute is mapped to an owl:datatypeProperty; a Code lists in ISO is mapped to a class in OWL; and the elements of the list are presented as instances of an owl class. In

ISO the values an element can take are said to be the domain of the element, while in OWL is said to be the range of a property.

Table 8. ISO elements definitions mapped to OWL

ISO	OWL
Name	rdfs:label
Short Name	rdfs:label
Definition	rdfs:comment
Obligation/Condition (Mandatory, Optional and Conditional)	Only mandatory implemented : owl:minCardinality=1 or owl:Cardinality =1
Maximum Occurrence	owl:maxCardinality
Data type	See Table 9
Domain	rdfs:range
Range of a data type property	rdfs:range rdf:resource= "http://www.w3.org/2001/XMLSchema# *datatype*"
Range of an object Property	rdfs:range rdf:resource="#name_of_class"

Table 9. ISO Types mapped to OWL

ISO	OWL
Class	owl:Class
codeList	owl:Class
Enumeration	owl:Class
abstractClass	owl:Class
codeListElement	instance of a owl:Class
DataType (user defined)	owl:Class
DataType (primitive)	xsd:*type*
characterString	xsd:String
Integer	xsd:int
Date	xsd:date

8.2 FGDC in OWL

The FGDC-STD-001-1998 metadata model is expressed as a hierarchy of compound elements. Compound elements contain other compound elements or data elements. A data element is the primitive unit and has a defined data type and a domain. The data type can be an integer, a real, a text, a date or a time. The domain can be a list of restricted and unrestricted values, or a free value depending on the type (e.g. free data, free integer or free text).

FGDC-STD-001-1998 does not conform to any conceptual formal specification such as an Object Oriented Model (like UML) or an Entity Relational (ER) model. This lack of conceptualization leads to different possible approaches to formalize FGDC in OWL. The two possible non-exclusive ways are:

1. Creation of arbitrary object properties to link elements seen as classes.
2. Creation of arbitrary classes to categorize elements seen as properties.

An example of case 1 can be found in http://loki.cae.drexel.edu/~wbs/ontology/list.htm and an example of the second approach can be found in http://loki.cae.drexel.edu:8080/~how/pangloss/. The two cases will be presented for the declaration in FGDC of the complex element *Keywords*:

Keywords = 1{Theme}n + 0{Place}n + 0{Stratum}n + 0{Temporal}n

The previous statement means that *Keywords* is a complex element that contains minimum one *Theme* and *Theme* can appear various times (maximum =

n), and Place, Stratum and Temporal elements are optional (minimum=0) and can appear various times (maximum = n).

8.2.1 Case 1: Creation of arbitrary object properties

The first possible way to conceptualize the previous statements is to view all the elements as classes and create arbitrary object properties. So the following is created for the Keywords statement:

Keywords = class

hasTheme = ObjectProperty, whose range is Theme

hasPlace= ObjectProperty, whose range is Place

hasStratum= ObjectProperty, whose range is Stratum

hasTemporal= ObjectProperty, whose range is Temporal

hasTheme, *hasPlace*, *hasStratum* and *hasTemporal* are all arbitrary object properties, whose ranges are *Theme*, *Place Stratum* and *Temporal* respectively. These ranges are complex elements seen as classes. For example, for *Theme*, the following is stated in the FGDC standard:

Theme = Theme_Keyword_Thesaurus + 1{Theme_Keyword}n

And to conceptualized the previous statement the following is done:

Theme = class

hasTheme_Keyword_Thesaurus = datatypeProperty ,

whose range is String

hasTheme_Keyword = datatypeProperty ,

whose range String

Theme is a class, and two arbitrary datatypeProperties are created: *hasTheme_Keyword_Thesaurus* and *hasTheme_Keyword*. They are datatypeProperties, because they correspond to simple elements in the FGDC declaration. And the range of both properties are xsd:String.

A conclusion for case 1 is as follows: For each element create a *has_element* object Property, if the range is a complex element, otherwise (i.e. range is a simple element) create a datatypeProperty.

8.2.2 Case 2: Creation of arbitrary classes

Case 2 differs from case 1, because in case 2 arbitrary classes are created: *Theme_Type*, *Place_Type*, *Stratum_Type* and *Temporal_Type*. The exception is the root, *Metadata*, which is seen as a class, and not as a property. This approach is similar to the serialization of FGDC in XML schema [104], where element types are created (i.e. similar to classes) to related complex element to contained elements. Classes and object properties are created as follows:

Keywords = class

Theme = ObjectProperty, whose range is Theme_Type

Place= ObjectProperty, whose range is Place_Type

Stratum= ObjectProperty, whose range is Stratum_Type

Temporal= ObjectProperty, whose range is Temporal_Type

Theme_Type = Class

Theme_Keyword_Thesaurus = datatypeProperty,

whose range is String

Theme_Keyword = datatypeProperty,

whose range is String

For case 1 and case 2, cardinalities and code lists are treated in the same fashion as ISO [64]. The second approach is preferred over the first one because creation of classes can be viewed as the process of categorizing properties, and secondly because a related approach exists in XML schema facilitating the conversion of one format to another.

This chapter presented strategies to conceptualize ISO and FGDC metadata standards into OWL. ISO original formalization is in UML, which is use for constructing object oriented models. UML to OWL conversion is preformed by matching UML concepts with OWL concepts and converting datatypes from one schema to the other. For the creation of the FGDC ontology it was required to create arbitrary classes, which can be seen as a mechanism to categorize FGDC properties.

CHAPTER 9: CONCEPTUALIZING CONTROLLED VOCABULARIES

One of the main problems with metadata specifications is the lack of domain-specific elements. To facilitate the use of domain terms with metadata specifications, this thesis proposes that both, metadata specifications and controlled vocabularies need to be conceptualized. The conceptualization should be expressed in a logic that allows to connect one with the other in a flexible and simple fashion. The connectivity is possible in Semantic Web Languages such as RDF and OWL.

The idea to present controlled vocabularies in an ontology form is not new. A recent study [75] shows that RDF has being used in various systems to encode dictionaries and vocabularies. And [79, 111], shows the use of ontologies to encode domain vocabularies; however for Hydrology very little has been done so far.

This chapter presents a taxonomy of controlled vocabularies, a survey of related vocabularies to the hydrologic domain and recommended methodologies to formalize controlled vocabularies in ontologies in order to facilitate the bridging between metadata specifications. First, a definition and classification of controlled vocabularies are given, followed by examples for the hydrologic domain, a methodology and finally some conversion examples.

9.1 Controlled vocabularies

A control vocabulary can be defined as a set of restricted words, used by an information community when describing resources or discovering data. An information community uses a controlled vocabulary to avoid misspellings and avoid use of arbitrary words that cause inconsistencies when cataloging data.

A consensus about what exactly constitutes a controlled vocabulary and the types that exist are both not clear. Ontologies can be viewed as controlled vocabularies [67, 79], while a controlled vocabulary can also be presented as a simple terminological tool [81]. Controlled vocabularies in digital libraries are seen as a Knowledge Organization System. Hodge [57], categorized these systems based on structure and complexity, relationships among terms, and historical functions, as follows:

- Term Lists:
    - Authority files: List of terms with limited hierarchical structures.
    - Glossaries: List of terms usually with definitions, capturing specific domain terms.
    - Dictionaries: General glossaries, with no hierarchical structure, which provide additional information about the term (e.g. origin, synonym, or related term).
    - Gazetteers: List of place names, usually classified or categorized.
- Classification and categories:
    - Subject headings: Shallow hierarchical structure of general terms.
    - Classification schemes, sometimes called taxonomies or categorization schemes which provide a hierarchical structure based on categories.
- Relationship Lists
    - Thesauri: Relationship list with more expression power than alphanumeric classification systems, because they can relate terms

by saying that one term is broader, narrower, related, synonym or antonym.

- Semantic Networks: Relationship lists that include thesauri relations plus other relations, such as whole-part, cause-effect, or parent-child relationships.
- Ontologies: Tools to represent complex relationships among terms, include rules and axioms not possible to express in semantic networks.

9.2 Classification of controlled vocabularies

The different Knowledge Organization Systems presented by Hodge [57] are classified in Figure 38 based on the level of conceptualization and categorization.

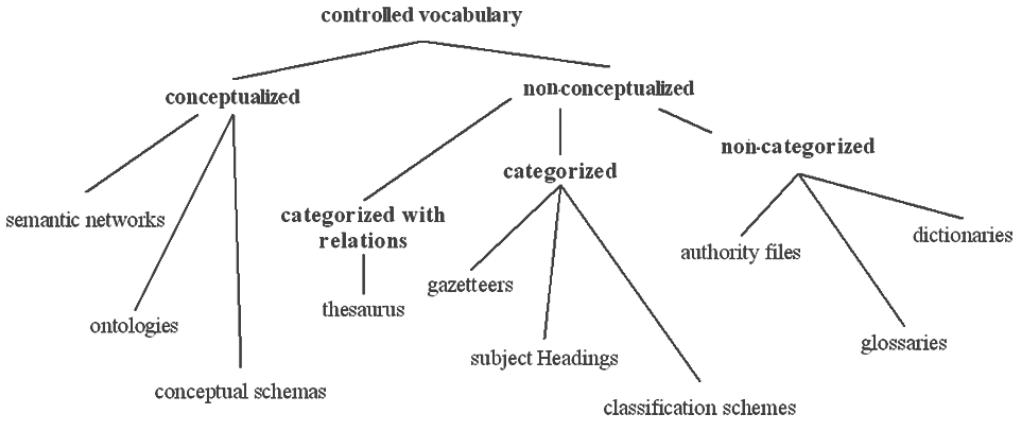


Figure 38. Classification of Hodge knowledge organization systems.

Controlled vocabularies can be characterized by differences in the conceptualization status. A controlled vocabulary can be *conceptualized* or *non conceptualized*. Three distinctions are depicted among non-conceptualized vocabularies: *Categorized*, *non-categorized* and *categorized with relations*. These categories are explained in detail in this section. Also, in addition to Hodge organization systems, conceptual models are added as conceptualized controlled vocabularies.

9.3 Definitions

An *abstraction* is a process where real world objects, their relationships and the properties of those objects are identified. The term *class* will be used to denote a general reality abstraction, as it is used in Object Oriented Languages and Knowledge Representation Languages. However, this term also refers to *entities* in Entity Relational Diagrams or *facets* in frame-based logics. A *formal conceptual language* (e.g. UML, RDF, OWL, ERD, FRL) is a system of written symbols and a grammar rules used as mechanism of communication between people and computer programs, that allows to express classes, their relations and their properties. A *symbol* is a conventional sign like box in UML or a tag in RDF/XML that explicitly communicates that the term associated with the symbol is a class. A class acts like a flag to tell if the controlled vocabulary is conceptualized or not.

9.3.1 Definition of conceptualized controlled vocabularies

A Controlled Vocabulary(CV) is conceptualized if:

1. A *symbol* explicitly exists to represent *abstraction* (e.g. class or box, tag).

2. At least one term uses such symbol to represent a conceptualization abstraction. (e.g. the term inside a box that denotes a class).

An example of a conceptualized controlled vocabularies for the hydrologic domain is ArcHydro (Maidment, 2002), which is a successful data model for hydrology and hydrography [77] another example is SWEET [103] which presents a conceptualized view of the Global Change Master Directory [95].

9.3.2 Definition of non conceptualized CV

A CV is non conceptualized if a symbol that denotes a class does not exist. Three types can be named: categorized, non-categorized and categorized with relations.

9.3.3 Non conceptualized categorized CV

A controlled vocabulary is non-conceptualized and categorized if it meets the following conditions:

1. It is non conceptualized.
2. At least one explicit category exists.
3. At least one term is classified.

An example of non conceptualized but categorized control vocabulary is a gazetteer. Two well known gazetteers are Getty Geographical Thesaurus names [41] and the Geographic Names Information System-GNIS [128]. GNIS contains about 2 million geographic features in the United States with name, latitude and longitude coordinates for each feature.

9.3.4 Non conceptualized - Non categorized CV

A controlled vocabulary is non conceptualized and non categorized if:

1. It is non conceptualized.
2. No explicit category exists.

Non categorized CVs include glossaries, dictionaries and authority files. An example of the later is a list of gauge stations of a basin. Relevant glossaries and dictionaries related to hydrology are the following: UNESCO International Glossary of Hydrology [127], which presents a list of words and their definition and translation in different languages, with “see also” links to similar terms. Water Science Glossary of Terms [86] developed by Nevada Division of Water Planning, which is a small internet version of a more complete dictionary, the Water Words Dictionary published by Nevada Division of Water Resources [59]. This last one is the more comprehensive glossary (about 386 pages). And a final example is a limited glossary use for education purposes developed by Oklahoma Climatological Survey [94].

9.3.5 Non conceptualized - categorized with relations CV

A controlled vocabulary is non conceptualized and categorized with relations if:

1. It is non conceptualized.
2. An explicit category exists.
3. Terms are related via means different than the categorization (e.g., *use for*, is *related to*).

This type of controlled vocabularies includes classification schemes and thesaurus. The most important cataloging systems are the Dewey Decimal

Classification [92], the Universal Decimal Classification [125] and The Library of Congress [120]. These are called also alphanumeric classifications because it uses alphanumeric codes to organize the terms from more general to more specific [81]. For the three systems, the place in the classification where the word *hydrology* appears is shown bellow:

Dewey Decimal Classification:

- 550 - Earth Sciences
- 551 - Geology, hydrology, meteorology
    - 551.4 Geomorphology and hydrosphere, water
        - 551.46 Oceanography
        - 551.48 Hydrology
        - 551.49 Ground Water Hydrology
    - 551.5 Meteorology
    - 551.6 Climatology and Weather

Universal Decimal Classification :

- 5 Natural science
    - 55 Earth sciences. Geology, meteorology, etc.
        - 551 General geology. Meteorology. Climatology. Historical geology. Stratigraphy. Palaeogeography
            - 551.1/.4 General geology
            - 551.1 General structure of the Earth
            - 551.2 Internal geodynamics (endogenous processes)
            - 551.3 External geodynamics (exogenous processes)
            - 551.4 Geomorphology. Study of the Earth's physical forms
            - 551.5 Meteorology
        - 556 Hydrosphere. Water in general. Hydrology

The Library of Congress :

GB651 – 2998 Hydrology. Water

GB980 - 2998 Ground and surface waters

GB980-992	Watersheds. Runoff. Drainage
GB1001-1199.8	Groundwater. Hydrogeology
GB1201-1598	Rivers. Stream measurements
GB1601-2398	Lakes. Limnology. Ponds. Lagoons
GB2401-2598	Ice. Glaciers. Ice Sheets. Sea Ice
GB2601-2798	Snow. Snow surveys
GB2801-2998	Hydrometeorology

GC1-1581 Oceanography

Well known thesauri related to hydrology are: NASA, Global Change Master Directory [95], which provides a list of keywords where category is the *broader* term, followed by *topic*, *term* and *variable*; the CIESIN Indexing Vocabulary [110] which is divided in science topics with “Land and Freshwater resources” being the one most related to Hydrology and also presents the words in a topic-term-variable hierarchy; the Alexandria Digital Library Feature Type Thesaurus [121] which contains a comprehensive list of feature types, and relations “used for”, “related terms”, “narrower term” and broader “term”; and the Florida Environments Online Thesaurus, developed by a cooperative initiative of the public universities of Florida, Publication of Archival, Library & Museum Materials [101], which presents broader and narrower, and *use for* relations.

9.4 Methodology

Based on the conceptualization taxonomy of controlled vocabularies a methodology is presented to conceptualize controlled vocabularies, starting from simple CVs to conceptualized CVs. A simple CV is one that has no

conceptualizations and no categories. The first step is to find a category that can group all the terms. Sometimes an obvious category can be found, such as “station” for a list of stations, or “science keywords” for a list of scientific terms. The issue is to select a class that can take on the role of a classifier for all the terms. This is a classical bottom-up approach, where common properties in real world objects are identified and are used as classifiers (i.e. classification abstraction [5]). For example, real world objects, USA, Colombia, and Germany have in common that they are a territory of a recognized nation. The two properties, territory and recognized nation promote a new class named Country. If a category exists in a CV, it is automatically converted to a class and as in the previous case, all the other classified terms are converted to instances of these class.

Conversion of a CV with categories and relations, such as thesauri and alphanumeric schemes, presents a greater challenge. A method of conversion of a thesaurus to RDF/OWL is explained in Assem et al., (2004). Basically the process is broken into two parts: first a syntactic conversion to a plain RDF and then a semantic conversion to OWL. Assem et al., (2004) present two conversion examples: Medical Subject Headings [124] and Wordnet, [31]. Another example of conversion of a thesaurus to an ontology is presented in [44], In this work the National Cancer Institute Thesaurus was converted into classes and broader classes were created to group the terms.

In general the following guidelines should apply:

1. Create classes, using preferred terms as RDF labels. Classes are created from concepts or categories that exist explicitly or implicitly.

2. Relations are converted to properties. But some properties declared in RDF schema and in OWL can be used (e.g. For example, *comment*).
3. Narrower-Broader relationships are mapped to object properties and are transitive and inverse.
4. Noun, verb, adverb, and adjectives are all disjoint.
5. Hyponym (subclass) and hypernym (superclass) can be converted to a new property or the subclass relation in OWL can be used. It is important to notice that superclasses can not be declared in OWL.
6. Synonyms can be mapped using sameAs relationship in OWL.

Two conversion examples are shown. One for a classification scheme and another one for a thesaurus.

9.4.1 Hydrologic Units Ontology

An example of a classification scheme in hydrologic vocabularies is the USGS Hydrologic Units Code (HUC) System, which is a hierarchical classification of nested large-to-smaller watersheds within a certain region. A HUC for the first level is formed by two digits (e.g. Mid Atlantic Region is 02) , for the second level is formed by four digits (e.g. Delaware Subregion is 0204), for the third level is formed by six digits (e.g. Lower Delaware Accounting Unit is 020402) and for the fourth level is formed by eight digits (e.g.Schuylkill Cataloging Unit is 020402).

Four categories can be found in the USGS classification: Region, Subregion, Accounting Unit and Cataloging Unit. All of these are mapped to classes (See Figure 39). Also a more general classification can be created:

Hydrologic Unit, which is also mapped to a class and becomes a super class of the previous four classifications.

Hydrologic Units are related via whole-part relationships. The property “is part of” is used for this purpose. OWL allows to express this property as a transitive property. For this reason it is not necessary to express explicitly that Schuylkill is part of the Delaware Subregion. Saying that the Schuylkill is part of the Lower Delaware and that the Lower Delaware is part of Delaware Subregion is sufficient.

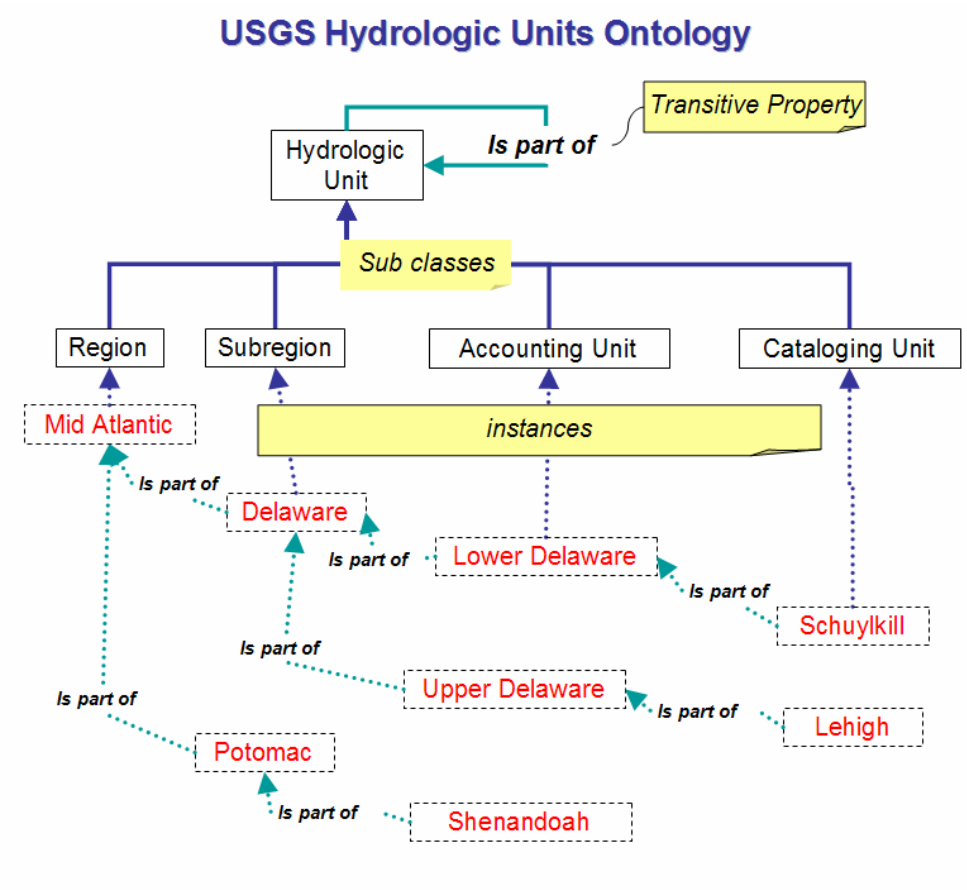


Figure 39. USGS Hydrologic Units Ontology

9.4.2 Global Change Master Directory Ontology

The Global Change Master Directory, GCMD, [95] provides a list of comprehensive keywords related to science. (An excerpt is shown in Figure 40). GCMD is organized in hierarchical way from broader to narrower terms (i.e. categories, topics terms and variables).

CATEGORY > TOPIC > TERM > VARIABLE

EARTH SCIENCE > Atmosphere > Precipitation > Rain
EARTH SCIENCE > Atmosphere > Precipitation > Sleet
EARTH SCIENCE > Atmosphere > Precipitation > Snow
EARTH SCIENCE > Cryosphere > Snow/Ice > Snow Density
EARTH SCIENCE > Cryosphere > Snow/Ice > Snow Depth
EARTH SCIENCE > Human Dimensions > Natural Hazards > Fires
EARTH SCIENCE > Hydrosphere > Ground Water > Saltwater Intrusion
EARTH SCIENCE > Hydrosphere > Ground Water > Springs
EARTH SCIENCE > Hydrosphere > Ground Water > Water Table
EARTH SCIENCE > Hydrosphere > Surface Water > Runoff
EARTH SCIENCE > Hydrosphere > Surface Water > Stage Height
EARTH SCIENCE > Hydrosphere > Surface Water > Rivers/Streams

Figure 40 Excerpt Global Change Master Directory Keywords.

Conversion from GCMD list to OWL is achieved by: 1) Presenting all keywords as classes; 2) Presenting all keywords as instances, 3) A combination of the previous two.

All GCMD keywords can be encoded as classes, assuming that the keywords have a clear hierarchical structure. *Rain*, *precipitation*, *atmosphere* and *EARTH SCIENCE* are classes. *Rain* is subclass of *precipitation*, *precipitation* is a subclass of *atmosphere* and so on. This approach is similar to the classification for

geologic ages and rock types [72] and the OWL encoding of the National Cancer's Thesaurus [44]. The problem with this approach is that the relations that exist among the classes and subclasses does not conform to common generalization abstractions [5], where "isA" relation exist among the classes and subclasses. For example, *precipitation isA atmosphere* is not true.

The second approach presents keywords as instances while *Category*, *Topic*, *Term* and *Variable* are represented as classes. A *Keyword* class is created as a parent of *Category*, *Topic*, *Term* and *Variable* classes. This translation preserves the organization of GCMD keywords, and conforms to classification abstractions [5]. For example, *Rivers/Streams* are GCMD variables, and *Hydrosphere* is a GCMD topic. In order to link the keywords it is necessary to define a property *hasParentCategory* in the *Keyword* class. For each subclass, restrictions are created on the *hasParentCategory* property to maintain the exact GCMD keyword structure. For example, *Category* does not have parent categories, so it has a cardinality restriction that says that *hasParentCategory* = 0. The *Variable* class, has only one parent category, *Term*, so it has a restriction for *hasParentCategory,* which *hasAllValuesFrom* the class *Term.*

The third approach is a combination of the previous two, which purpose is to facilitate the use of GCMD keywords in metadata specifications. For example, a restriction can be created on *iso:keyword* (see Figure 41), so that *iso:keyword* can take only values of selected GCMD terms (e.g. *Atmosphere*, *Climate indicators*, *Cryosphere*, *Land Surface* and *Oceans*).

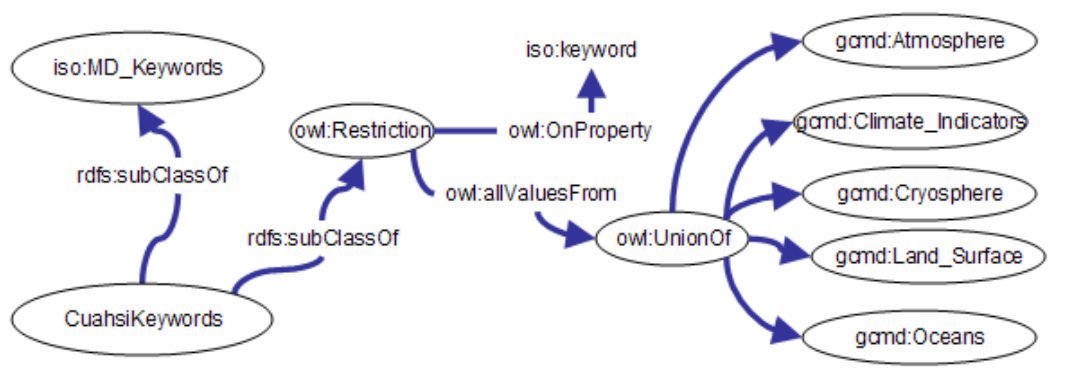


Figure 41. Extension of ISO MD_Keywords to accept all values of GCMD

In order to create the previous restriction, two conditions are necessary: 1) that the keywords are individuals of an existing class and 2) that *Atmosphere*, *Climate_indicators*, *Cryosphere*, *Land_Surface* and *Oceans* are declared classes. For these reasons, all the GCMD variables were encoded as instances and classified under their parent term-topic-category. A Protégé snapshot of such conceptualization is presented in Figure 42.

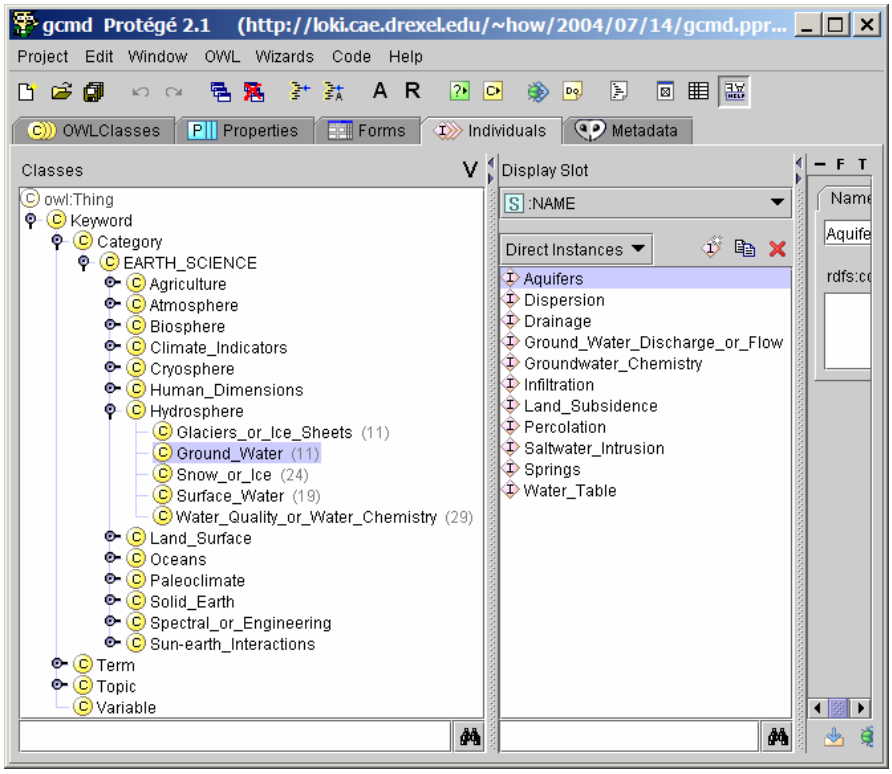


Figure 42. Protégé snapshot GCMD in OWL

This chapter presented a definition of controlled vocabularies and a categorization of Hodge's Knowledge Organization System. A strategy to conceptualize control vocabularies depending of the level of conceptualization was also discussed. Two complete examples were studied: a hydrologic units ontology and the Global Change Master Directory ontology.

CHAPTER 10: UPPER HYDROLOGIC ONTOLOGY

10.1 Motivation and background

The previously discussed controlled vocabularies in Chapter 8 are not always available in machine-readable format. If any of these are to be used in the Semantic Web they must be converted into a conceptualized schema. Larger collections of terms are found in general thesaurus such as Wordnet [31]; “the most widely used ontology for natural language processing”[114]. However, Wordnet, which can be found in RDF [80] and OWL [69] lacks of specialized hydrologic concepts. The incompleteness of the controlled vocabularies previously mentioned will be demonstrated in the following examples:

- Wordnet does not contain terms like “specific yield” or “NEXRAD”,
- International Glossary of Hydrology [127], does not define “atmosphere” or “geological formation”.
- Global Change Master Directory (GCMD), does not contain hydrologic terms like “well”, “transmissivity”, or “dikes”.
- Alexandria Digital Library Feature Type Thesaurus being the most complete feature thesaurus, does not contain “gauge station” or “pollutant discharge source” or related terms that can be used instead.
- The Water Words Dictionary published by the Nevada Division of Water Resources does not contain terms to describe *geologic formations*.

For the hydrologic domain, there is a lack of top level ontology that can help categorized new terms. This chapter presents such an ontology, where existing terms from other ontologies as well as new terms can find a place in a hydrologic

conceptualization. The purpose is to create an ontology using a top-down approach, creating a conceptual model to fit hydrology related data.

An upper hydrologic ontology is a controlled vocabulary that defines classes to help categorize terms related to the hydrologic domain. The classes are related to each other via properties. Some properties have logical characteristics, allowing agents or computer programs to infer related terms and provide a better query result to users or service requesters. The upper hydrologic ontology can be used for:

- inference of hydrologic knowledge (further explained in section 10.4), and
- refinement of hydrologic related searches (further explained in section 10.4).

The upper hydrologic ontology can also served as:

a controlled vocabulary when creating or searching hydrologic metadata, similar to what was created for GCMD (See

- Figure 41), and
- a central ontology to solve semantic heterogeneities occurring in other ontologies, by mapping other terms to the hydrologic top concepts.

10.2 Construction and definition of terms

The upper hydrologic ontology has been built around the concept of measurement. Measurement and the related concepts presented in Figure 43 are explained in detail in this section.

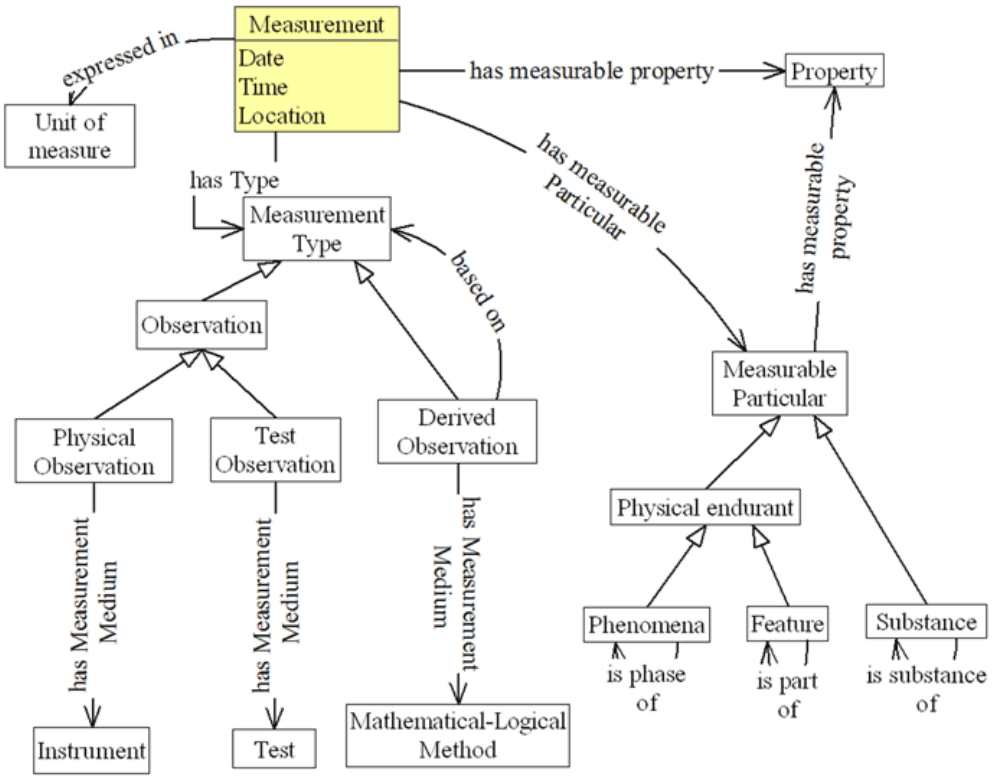


Figure 43. Upper Hydrologic Ontology

A ***measurement*** can be defined as "a process of assigning numbers or other symbols to an attribute of a thing, in such a way that the relationships of the numbers or symbols reflect relationships of the attribute being measured." [106]. Sometimes, *observation* is used as a more general term, to distinguish between *quantitative measurements* from *category observations* [38, 100]. From a classical point of view, measurements can be categorized as nominal, ordinal, interval and ratio [117]. In a nominal measurement objects are classified by shared attributes, which is the same as *category observations* [38, 100]. While the other three categories are quantitative measurements. Since the purpose of this thesis is to

present the semantics to facilitate interoperability, the divisions of *category observations* and *quantitative measurement* are not taken into account. The reason is that it is not envisioned that a user will search by “category measurement”. Instead he or she will rather search by an instrument (e,g, NEXRAD data) or test data (e.g. pumping test). If data is recorded by an observable (i.e. instrument or test) it is an observation, and if the measurement is a product of a mathematical or logical method it is said to be a derived observation.

An ***Observation*** is a measurement originating from an instrument or a test that can record a value or a symbol for a property of a phenomena. Observations are discrete samples of continuously running natural processes for which only one value exists in the spatial temporal domain for the measurement [85].

A ***derived observation*** is a measurement that is not observed but computed. In Wordnet *computation* is defined as a procedure of determining something by mathematical or logical methods. It is different from an observation because observation is not determined by mathematical or logical methods, but by an instrument. If *discharge* is calculated by a rating curve, it is a product of a computed value and not an observed value. Also, computing a runoff curve number by means of looking at a table of land use vs. hydrologic soil groups is a logical computation (e.g. if land is ”dirty road” and hydrologic soil group is “B” then curve number is 82). Another example is reflectivity which is a measured value while rainfall rate is a computed value from reflectivity data. Also, permeability is another example of derived observation.

Property is a basic or essential attribute shared by all members of a class [31]. Rainfall intensity is a property of rainfall, so all rainfall events have to have

rainfall intensities. It behaves like an abstract object, having no location in time and space [48]. Where and when questions have no meaning. For example, where is the pH? Or when is the pH? A property can be observed or computed. (e.g. evaporation rate measured in millimeters/day). For this reason a class name “Measurable particular” is created, which is allowed to have “measurable properties”.

Measurable particular is a hydrologic particular, which has a *measurable Property*. It is divided in *Phenomena* and *Physical Endurant*. They have part-whole relations as defined by Winston [134]. The component-object relation (e.g. channel-stream), member-collection (e.g., stream-waterBody) and place-area (e.g., stream-basin) are defined as the property *is part of* in the Physical endurant class. The stuff-object relation (e.g., water-stream) is defined as the property *has substance component* in the *Substance* class and the phase-process relation (e.g. evaporation, hydrologic cycle) is defined as the property *hasPhase* in the *phenomena* class. These relations are transitive and have corresponding inverse properties (e,g., *is part of* and *has part*).

The *is part of* relation when the relation is applied to a feature is not strictly a *whole-part* relation, since most of the time geospatial features have fiat boundaries [113]. This means that the they do not posses a divisible bulk. For example, it is difficult to discern where a mountain starts, or what the real boundaries of a basin are. Also, a river can be part of a basin, but not completely and for the case of geological formations, the differentiation is even harder. The *is part of* relation is simplified as a relation where the containment part does not need

to be completely inside the whole. If at least one part is contained, then it is said to be *part of the* containment feature.

Phenomena: The American Heritage® Dictionary defines phenomena in the physical sense as “an observable event” (e.g. rainfall, infiltration and runoff). It is a subject of the observation to which values can be assigned [100]. In the context of this thesis, a phenomena is an event that can be observed by instruments, observed by tests, or derived from other observations. A measurement is not a simulation, since it is not a product directly or indirectly of an observation.

A phenomena exist if anyone of its properties can be measured by an instrument, be reproduced by a test or be computed from previous measurements. It can be raining very lightly for a short period of time, but if the reflective measured by the radar is not greater than 5 decibels (in rain mode), then it is not raining. In other words, a phenomena is defined through its ability to be detected by an instrument. Moreover, to record a natural phenomena precisely more than one device is necessary, each of them performing different measures. For example, radar and rain gauge measurements together, can be more accurate for estimating rainfall than each measurement alone. Also, a phenomena can be estimated by a computation, which includes mathematical models, lookup values in tables, or simple equations.

The second part of the phenomena definition says that it is an event. An event, is a particular (i.e. unrepeatable entity) that *occurs* or *happens* or *takes place* [50], commonly known as occurrent [48]. Guarino [48], explains that an occurrent is generated by a continuant (i.e. object) and that it cannot exist without one. (e.g.

evaporation is a phenomena, that can only exist if water and the sun both exist). Also events can be static or dynamic.

In order, to precisely define an event it is assumed that events cannot move, are not properties and are not facts. Event though we perceive that a hydrologic event (e.g. rainfall event) moves because different events are measured and compared. Events are not properties. Casati and Varzi [14], present a discussion about events that can be seen as properties and vice versa. They argue, for example, that the redness of an apple is a property, but also an event with spatial temporal locations. The redness of the apple is perceived in a coordinate x and an interval time t. These properties are called phased sortals [49]. The hydrologic events precipitation, infiltration and runoff, even though they are three distinct events, can be seen as properties or phased sortals of the water flow. However if we insist on the previous definition that a property cannot have a temporal and spatial locust, then it can be concluded that if precipitation has a spatial and temporal location it is not a property, and therefore must be phenomena.

Events are not facts [102, 130]. A rainfall event should not be confused with the fact that it rained. For an event to be an event it needs the spatial and temporal extent. A fact only refers to the event. For example, the fact that the Schuylkill River flooded is distinguished from the event that Schuylkill River flooded at time t. To be precise, if something seems like a fact but it has clearly a spatial-temporal location it will always be an event.

A ***Physical endurant*** is define in DOLCE, Descriptive Ontology for Linguistic and Cognitive [78] as a top level concept. It is a useful category to separate measurable particulars from phenomena. A physical endurant, is a

particular that can endure in time and can be feature or a substance. The difference between a feature and a substance is that feature has a spatial location, while substance does not. A substance is sometimes called stuff, and it can be composed of other substances (e.g. Hydrogen and Water) or can be composed of other features (e.g. water is part of streams).

Instrument: Wordnet defines an instrument as a device that requires a skill for proper use. An instrument in this thesis is a device that records observations. If the instrument is located in the spatial location where the measurement is taking place it is said to be in situ (i.e. confined to the site of origin), e.g. gauge station, if not it is said to be remote, e.g. radar. These two distinctions are important because in the first case, the location of the instrument can be used as the location for the phenomena, which is different from the second case.

10.3 Testing of the top hydrologic ontology

To test the proposed upper hydrologic ontology, instances depicting different type of data were created in Protégé [116].

10.3.1 Feature

The first instance created was *basin*. *Basin* is a feature, since it has a spatial location and has a continuous existence (i.e. physical endurant). Since it is a measurable particular it has measurable properties, such as curve number, precipitation distribution and Digital Elevation Model (DEM). These are measurable properties since they do not have a space and time location. Also, to be a property they should have a measurement medium. Curve number can be measured by a logical method, such as the SCS curve number method, using the soil type tables.

A basin can contain other features such as streams, lakes, waste water treatment facilities, and geological formations, which do not need to be totally contained in the basin. If any part of a geological formation is in the basin, it is said to be contained. It should be clear that no distinction is made on how these features are represented (e,g, point, line, polyline or volumes).

10.3.2 Phenomena

Precipitation is the first phenomenon created, which is an event. It occurs. Precipitation has properties, such as precipitation rate (intensity) and precipitation volume, which can be measured using various mechanisms: remote sensors like NEXRAD, on-site instruments like rain gauges or buckets. Since NEXRAD does not measure the precipitation rate directly, it is said to be a *derived observation*. A rain gauge measures rain depth or volume, hence it is not a derived observation, while the synthetic unit hydrograph is a method to reproduce a measurement in a watershed.

10.3.3 Substance

The first obvious substance to be created is water, which is part of a water body, aquifer or the atmosphere. Water does not have a physical location, because it does not have a specified latitude or longitude. It is composed of other substances (H and O), and it has measurable properties such as temperature and pH.

10.4 Usage example: Hydroogle

A hydrologic ontology can serve a number of different purposes. One of them is to provide the controlled vocabulary for a community, similar to what was presented with the hydrologic units ontology and the GCMD ontology in earlier chapters. Dynamic Community Profiles can use the terms define in the ontology as controlled vocabulary for selected elements in extended metadata specifications. In addition, hydrologic ontologies can help discover knowledge and refine queries to improve search results by reducing the information overload. In order to test this approach, an application called Hydroogle was developed, which follows the idea of “Google”, which is the most popular search engine today.

Hydroogle helps to discover hydrologic knowledge by presenting related terms, as shown in Figure 44. The term RAIN was searched and terms that contained RAIN were displayed in HTML with properties that were linked to other hydrologic concepts. For example, *rainfall* is a type of *precipitation*. And *rain bucket* is a type of *point measurement*. If a user wants to search for other *point measurements*, a click on *point measurements* will present some types of *point measurements* (e.g. *evaporation pan*, *rain bucket*, *rain gauge*, *piezometer*, etc..)

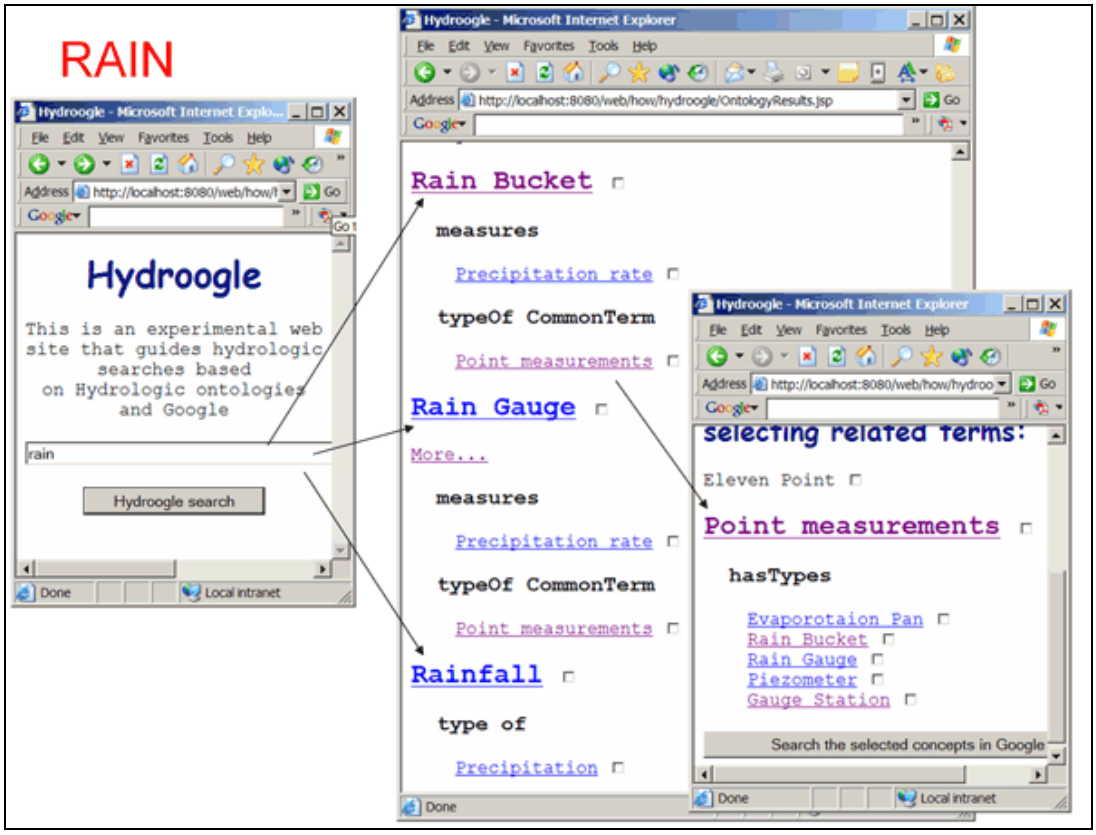


Figure 44. Discovering Hydrologic knowledge with Hydroogle

Hydroogle will also help to refine searches. In the first attempt to search for "Stage Delaware", Google retrieved 642,000 results. Google matches all the index Web pages that contain both *stage* and *Delaware*. The first results are related to festivals and entertainment information. Google is unable to discern that the search has a hydrologic context. The large amount of results found including non-relevant searches is what is commonly known as *information overload* [67].

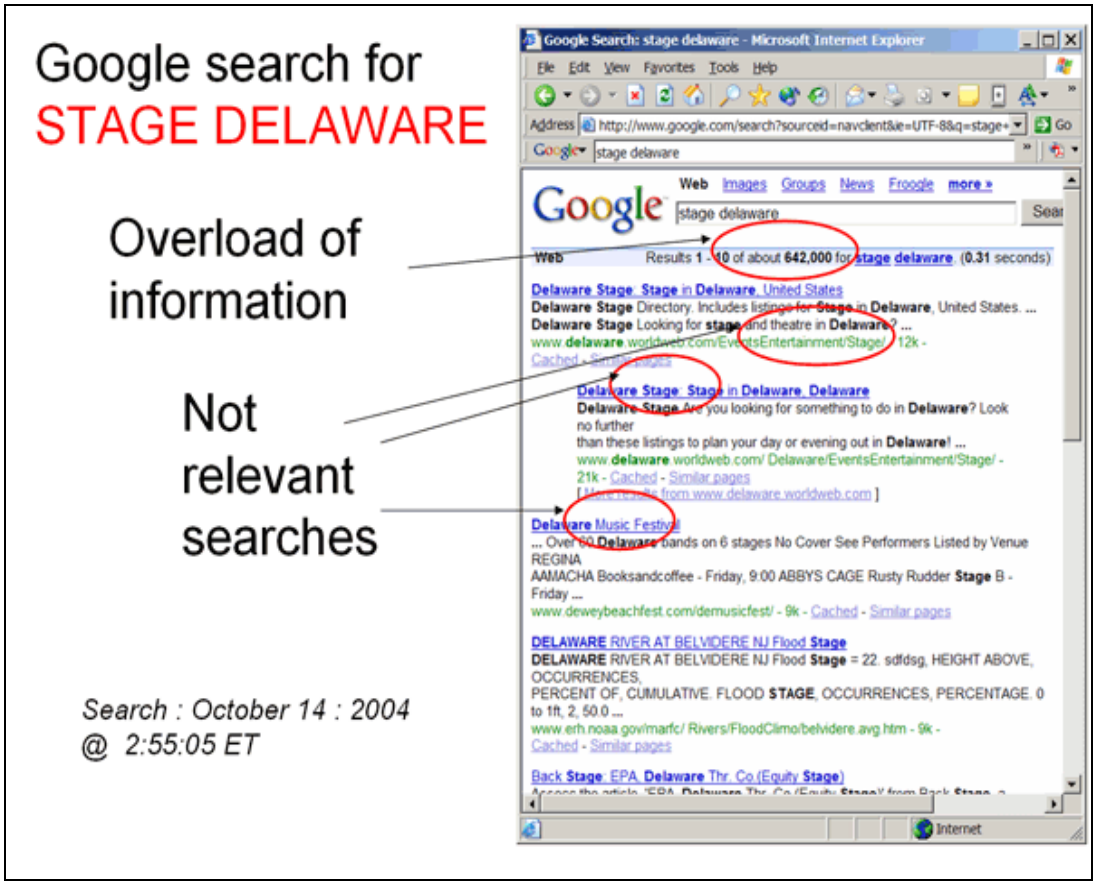


Figure 45. Google search for stage and Delaware.

To guide the user towards a more specific hydrologic search, Hydroogle presents a list of possible hydrologic units containing that name and the concepts related to any of the terms the user wants to search for. The user can select, for example, *Lower Delaware* and *stage* and *streamflow*, clicking on the small box at the right of the term. Then the Google Web Service is implemented to search for the refinement search. See Figure 46 and Figure 47.

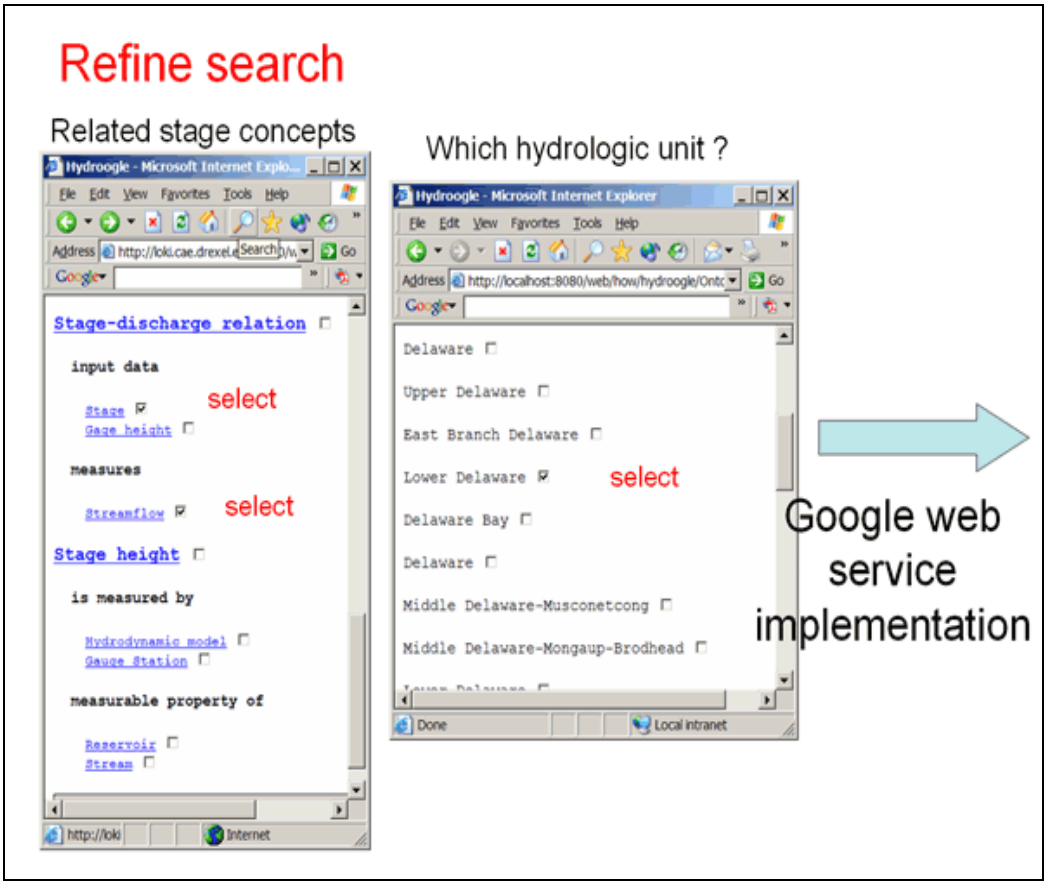


Figure 46. Search refinement Hydroogle

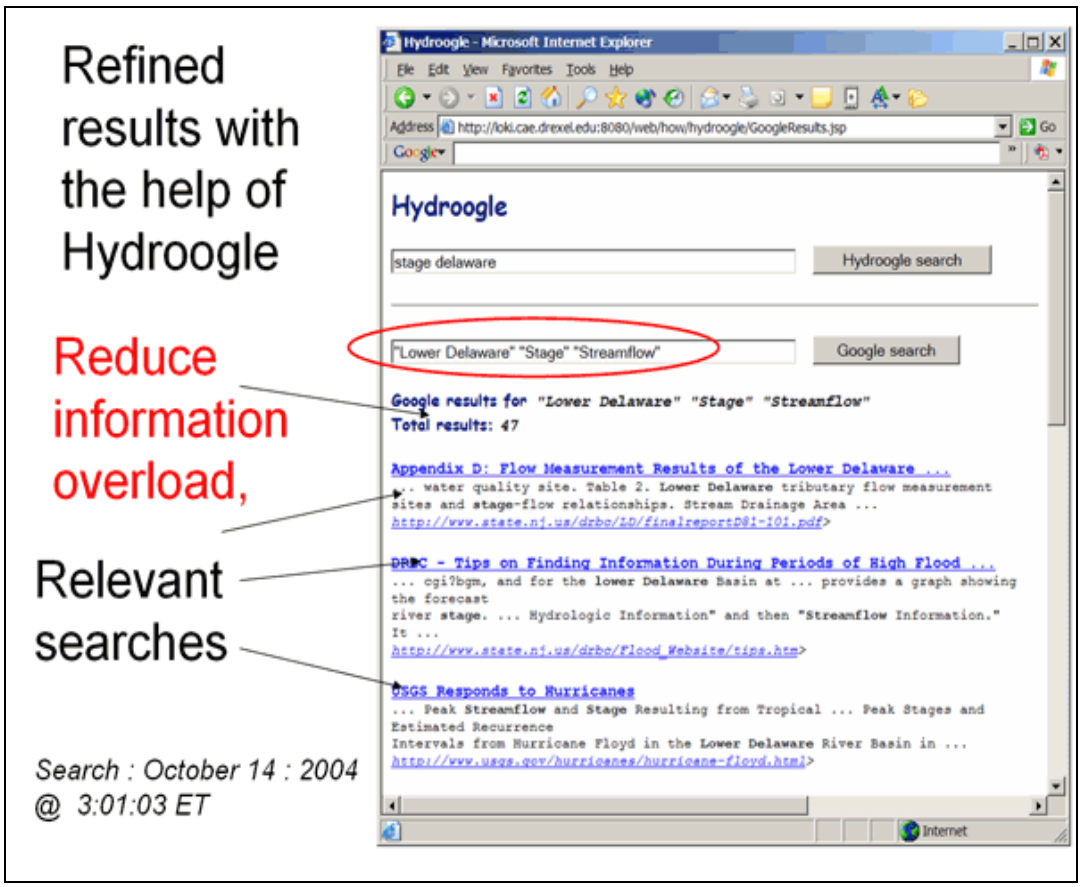


Figure 47. Refined search for stage an Delaware in Hydroogle

The architecture of the application is shown in Figure 48. The server contains two ontologies, namely the *hydrologic units* ontology and the *top hydrologic* ontology. It is composed of a *resource extractor*, which extracts the resource that match any of the user search terms if the resource is found, its properties and the range of its properties is presented in an HTML page. The user can then click a box behind the desired search term to be able to redefine the search. Once the user makes his selection, and clicks on the “Search the selected concepts in Google”, the *query orchestrator* creates a query that is sent to Google using the Google Web service.

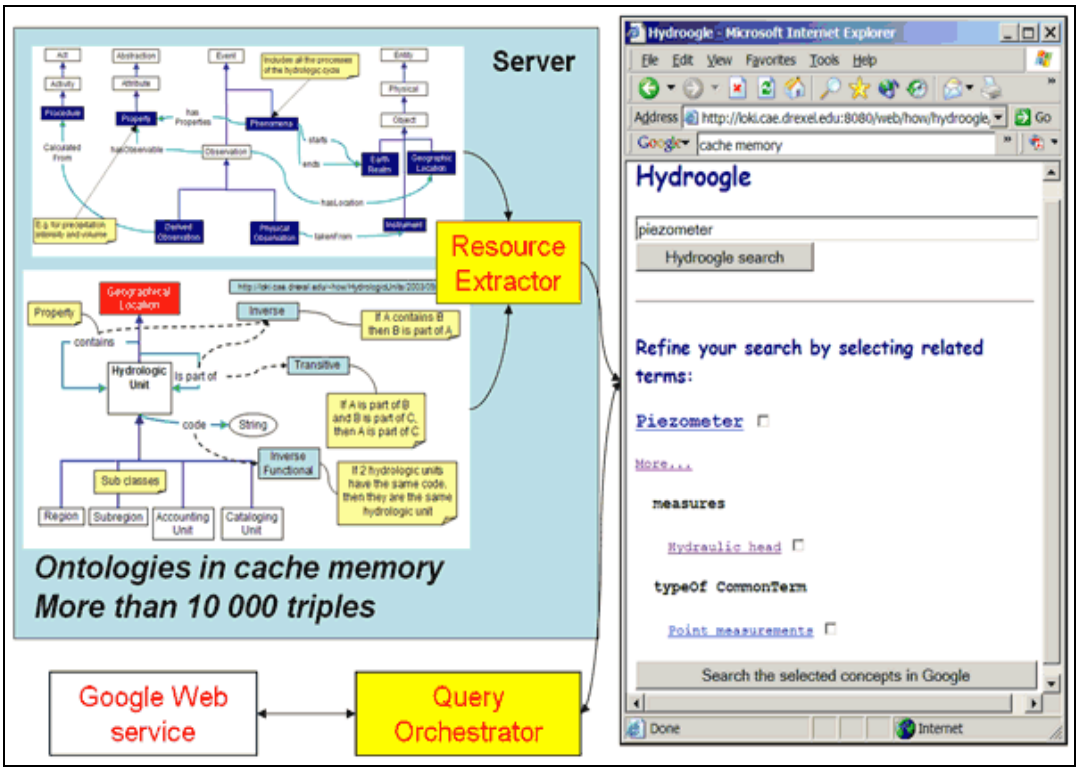


Figure 48. Hydroogle Architecture

This chapter presented a proposed upper hydrologic ontology, where hydrologic terms can be related together via top hydrologic concepts. An application, Hydroogle, was created using the proposed ontology, that helps to discover hydrologic knowledge and guides users towards a more specific hydrologic web search.

CHAPTER 11*: PANGLOSS*

11.1 Overview

To test ONTOMET, *Pangloss,* a suite of products that interact with metadata specifications and vocabularies, was developed. The prototype takes advantage of World-Wide-Web (WWW) recommended technologies, such as: Hypertext Transfer Protocol (HTTP) protocol to provide communication; Internet Protocol (IP) to provide connectivity across heterogeneous networks and hardware platforms; Universal Resource Locators (URLs) as a universal identifier and locator for distributed resource on the Web; Web Ontology language (OWL) and the Resource Description Framework (RDF) to represent metadata specifications, instances and vocabularies; eXtensible Markup language (XML) as the encoding medium and HTML to present metadata specifications.

A variety of programming languages (C++, Visual Basic, JAVA, Pyhton, etc.) exist to create a software program. However there are some advantages that JAVA has over the others. JAVA is license-free; its compilation units can be executed on any operating system, once the virtual machine has been installed in the computer; it is fully object oriented, facilitating reusing of the code and functional encapsulation; and lastly only JAVA APIs to interact with OWL were available (i.e. JENA [61]) in 2001. The three programs created were:

1. *Pangloss* MetaInstance, facilitates creation of instances based on schemas expressed in OWL ontologies and profiles created with the methodologies presented in this thesis. The result is an RDF/XML document that conforms

to the rules of the metadata specification. The screenshot (Figure 49) shows an example for ISO-19115:2003.

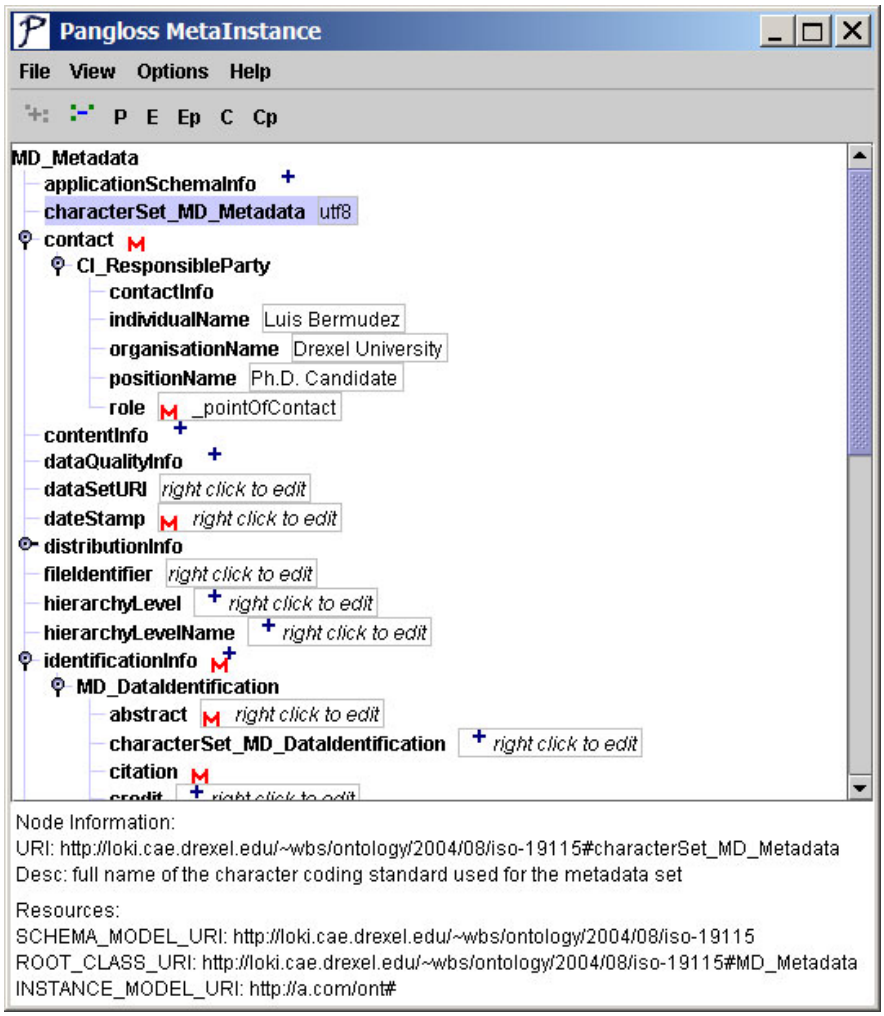


Figure 49. *Pangloss* MetaInstance

2. *Pangloss* MetaExtender permits the selection of core elements of a metadata specification expressed as an OWL ontology. These paths can then be used by an information community to set their profile. The screenshot (Figure 50) shows the core paths for ISO-19115:2003.

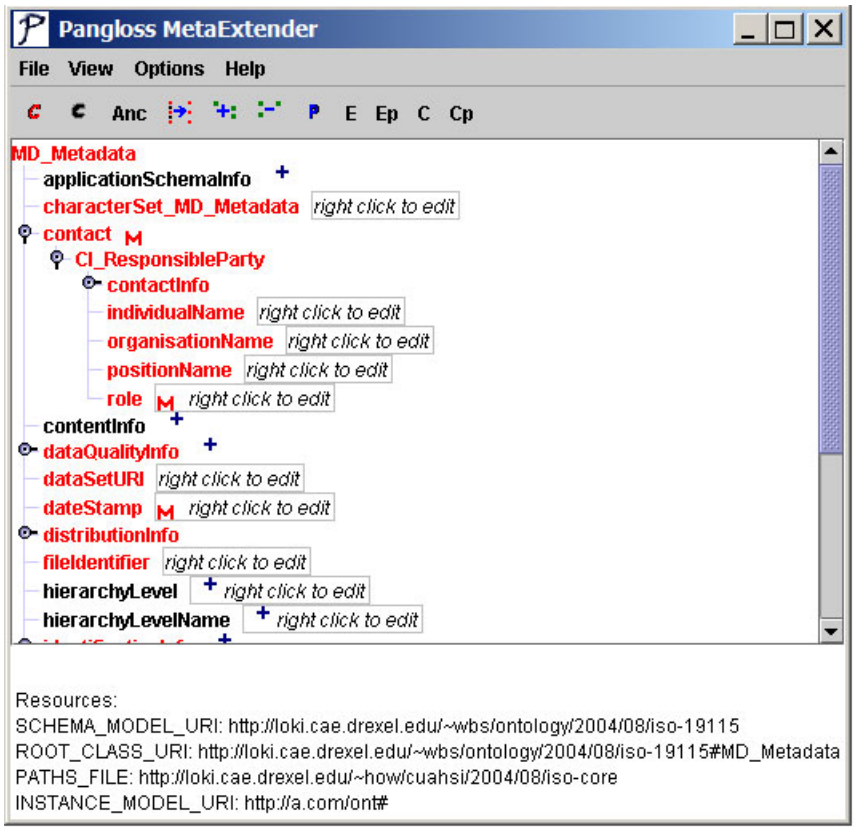


Figure 50. Core Paths of ISO-19115:2003 in *Pangloss*

3. *Pangloss* MetaMapper allows the explicit creation of a mapping between two metadata specifications, both of them expressed as an OWL ontology. The screenshot shows the mapping of FGDC Place-Keyword to ISO keyword + type=place. The model implemented follows the tree model approach.

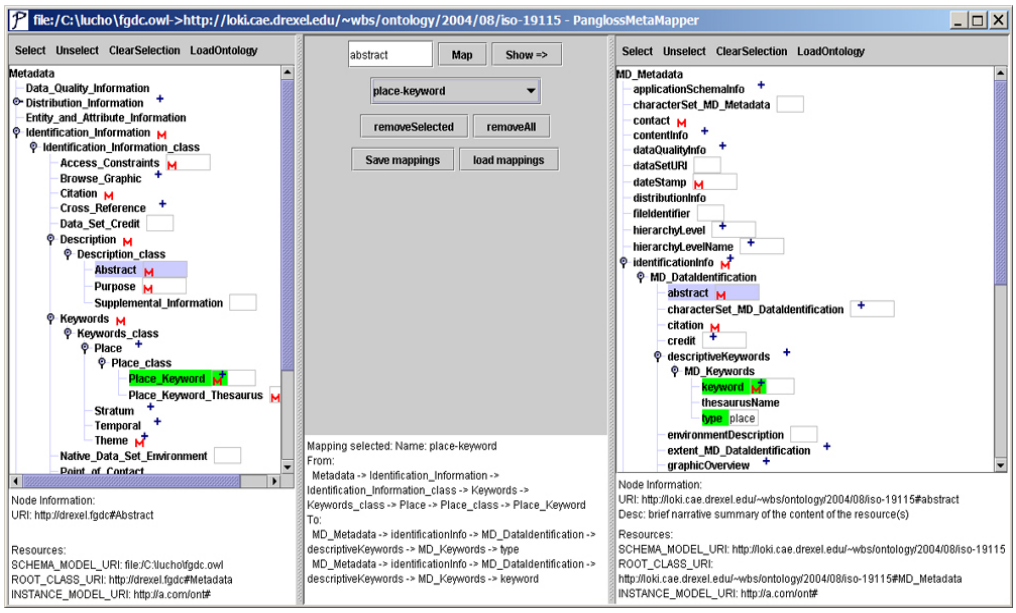


Figure 51. *Pangloss* MetaMapper

The main characteristics of *Pangloss* are :

- Flexibility: It can open any OWL schema (even those not related to metadata specifications). However, the root must be specified in order to create the class.property.class.property tree.
- Clarity:
    - Root class: It is clear where to start, since the root will display the first properties.
    - Mandatory elements: Mandatory elements are shown with a red M if the cardinality or the minimum cardinality is one.
    - Duplicable elements: If an element is duplicable a + sign is shown to indicate that the whole branch can be repeated

  - Datatype properties which need a value are presented with a blank blue square.

- Easy tree navigation. All the nodes in the tree can be navigated following the Class.property.class.property.. pattern. If a property has as range a class, the class and its subproperties are presented to the user.
- Loop problems are resolved by manual navigation. The class.property.class.property pattern can result in circular problems. *Pangloss* shows the first class and the first properties. After that the user is required to manually navigate along the tree, and to manually create loops. This does not affect the program performance.
- Fast loading. *Pangloss* opens two times faster than the best ontology editor (Protégé). This is due to the fact that, Protégé is based on its own knowledge-based model which is mapped to a JENA model, while *Pangloss* uses a JENA model directly. Also in *Pangloss* not all the classes and properties are open when loading the ontology, it only opens the root, and the first branch (properties), reducing loading time.
- Accepts class restrictions on datatype Properties, which is an OWL-Full expression. The encoding can be either of the two presented in Figure 52. The right one is preferred, where the URI is being encoded as a String. *Pangloss* checks for every string to see if it is a URI or not, and acts accordingly.

As object property:

```
<A rdf:ID="individual_a">
   <hasB rdf:resource="http://foo#b1"/>
</A>
```

As datatype property:

```
<A rdf:ID="individual_a">
   <hasB>http://foo#b1<hasB>
</A>
```

Figure 52. Datatype encoding of a URI.

- Restriction rules of inherited properties take precedence over parent properties as explained in section 2.3. This is useful to present the specification created form Dynamic Community Profiles methodology.

The use of this product to create a hydrologic metadata profiles is discussed in Chapter 12. A survey of other tools to annotate and extend metadata are presented in the next section.

11.2 Other tools to annotate and extend metadata specifications

The USGS provides a variety of tools to support creation and validation of metadata for CSDGM FGDC-STD-001-1998 in: http://geology.usgs.gov/tools/metadata/, including compilers, parsers, editors, online helpers to validate and create enumerated domains and source Information from bibliographic records. A metadata validation service for FGDC metadata is available at http://geo-nsdi.er.usgs.gov/validate.php. ESRI Arc Catalog facilitates and automates creation of metadata, while it is FGDC compliant it is not fully ISO compliant. The Environmental Information Management System [27], http://www.epa.gov/eims/index.html supports FGDC. It is a Web system that displays forms to populate metadata for EPA staff and other individuals by getting a

password, however it does not fully accommodate FGDC metadata standards and their output is only in HTML format. The California Environmental Information Catalog (CEIC), an online directory for reporting and discovery of information resources for California. http://gis.ca.gov/catalog/index.epl, allows users to create their own catalogs, but no standards are imposed. OMNIPORT formulates a way to create dynamic forms to parse metadata for distributed and heterogeneous environments [37]. The MORPHO tool available at http://knb.ecoinformatics.org/software/eml/ helps to create Ecological metadata based on EML, and MetaMaker helps to create FGDC compliant metadata. MOBE is a metadata editor developed by San Diego Super Computer Center. A simplified metadata schema, Metadata Transfer Format, MTF [55] is used as a template to create metadata. It is flexible in the sense that it accepts any MTF, and the syntax is simple; however, the following is not possible: expressing that a complex element has cardinality “many”, specifying a range of controlled vocabularies and create and edit RDF/XML or OWL documents.

Other tools which support FGDC, and ISO and support with relational databases are: Multistandard Multilingual Metadata catalogingTool (M3Cat) created by Intlec Geomatics, Enraemed Metadata Collector from the Ethiopian Natural Resources Environmental Meta-Database, Spatial Metadata Management System (SMMS) produced by Intergraph, and Metamanager created by Compusult Limited.

All of the previous tools, except MOBE and COBE have hard-coded the metadata elements, which makes them inflexible tools. While this has the benefit that it can sometimes be easier to create metadata, it has the disadvantage that

multiple profiles, if created following the methodology in this thesis, will need additional coding. Also, they do not allow editions in RDF/XML nor OWL.

The Protégé ontology editor, can be used as an alternative to create instances of metadata specifications expressed in OWL; however, treatment of core elements and class restrictions of datatype properties need special attention. Also, a tree view of a metadata specification as class.property.class.property is not possible in Protégé, which displays classes in one panel, and properties in another. Creating an instance for ISO:19115:2003 in OWL which has more than 100 classes is cumbersome in Protégé. For this reason, *Pangloss* presents a metadata tree view, similar to COBE, which is more user friendly when creating metadata instances.

This chapter presented *Pangloss,* a suite of products that interact with metadata specifications and vocabularies and that make use of the methodologies in ONTOMET. Also it presented differences with other tools that facilitate creation of metadata instances.

CHAPTER 12: CUAHSI METADATA PROFILE

The methodology to setup a Dynamic Community Profile was tested by creating a metadata community profile for CUAHSI. The requirements for the Drexel University Team, were to recommend a set of metadata elements to describe hydrologic data. The project had the following constraints:

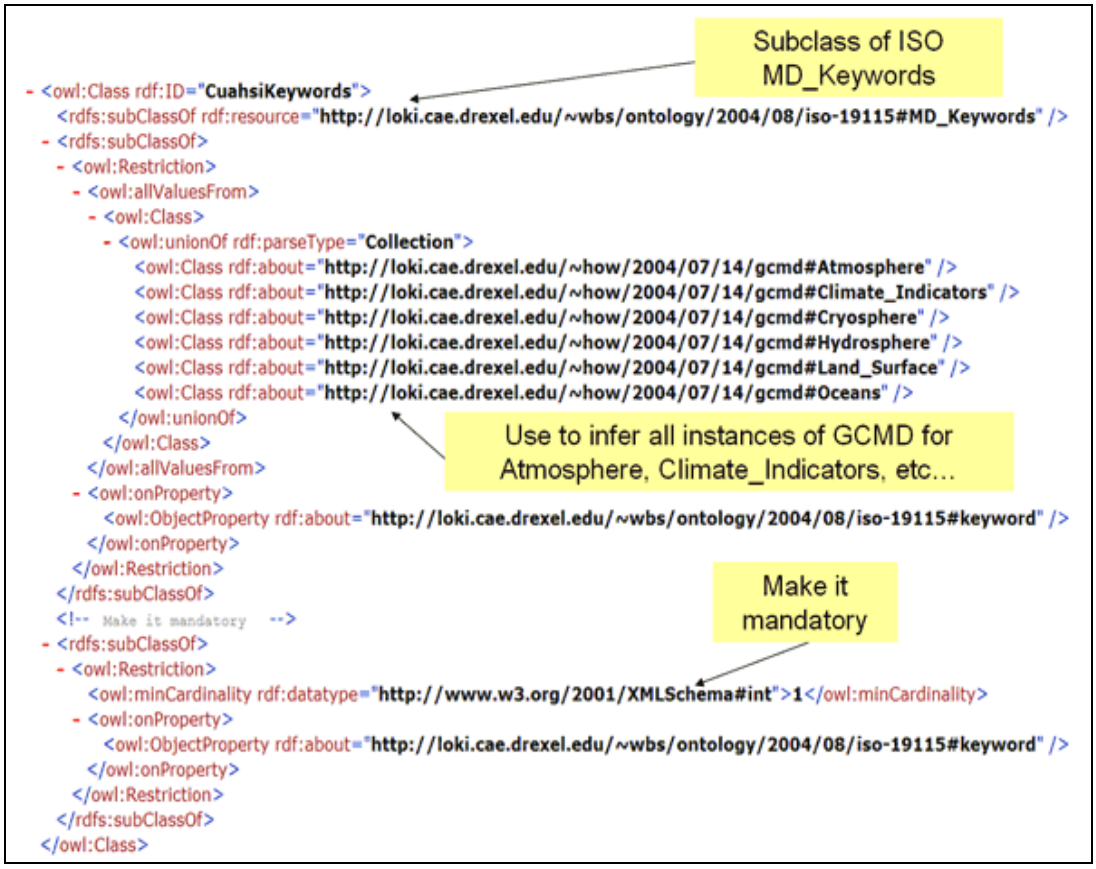
1. The elements should be published in Metadata Template Files (MTF) which is a special format preferred by the San Diego Supercomputer Center.
2. The elements must comply with the ISO:19115:2003 and related standards.
3. The profile must use controlled vocabularies from a known thesaurus for topic themes and terms from measurement units detailed in the Handbook of Hydrology [76]

The procedure was as follows:

1. The ISO:19115:2003 and related metadata specifications were converted to OWL, as outlined previously in section 8.1.
2. Controlled vocabularies for units and for topics were encoded in an ontology as outline in Chapter 9.

CuahsiKeywords was created as a subclass of iso:MD_Keywords. And a restriction was created on iso:keyword , so that iso:keyword can take only values of selected GCMD terms (e.g. Atmosphere, Climate indicators, Cryosphere, Land Surface and Oceans). See

3. Figure 41 for a graphical view and Figure 53 for the XML encoding.
4. CUAHSI Keywords were made mandatory.




```
- <owl:Class rdf:ID="CuahsiKeywords">
    <rdfs:subClassOf rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#MD_Keywords" />
  - <rdfs:subClassOf>
    - <owl:Restriction>
      - <owl:allValuesFrom>
        - <owl:Class>
          - <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Atmosphere" />
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Climate_Indicators" />
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Cryosphere" />
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Hydrosphere" />
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Land_Surface" />
              <owl:Class rdf:about="http://loki.cae.drexel.edu/~how/2004/07/14/gcmd#Oceans" />
            </owl:unionOf>
          </owl:Class>
        </owl:allValuesFrom>
      - <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#keyword" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <!-- Make it mandatory  -->
  - <rdfs:subClassOf>
    - <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      - <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#keyword" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

Figure 53. RDF/XML encoding of CUAHSI keywords creation.

5. The CUAHSI Metadata profile was created by extending ISO MD_Metadata, and importing the related ontologies, as shown in Figure 54.

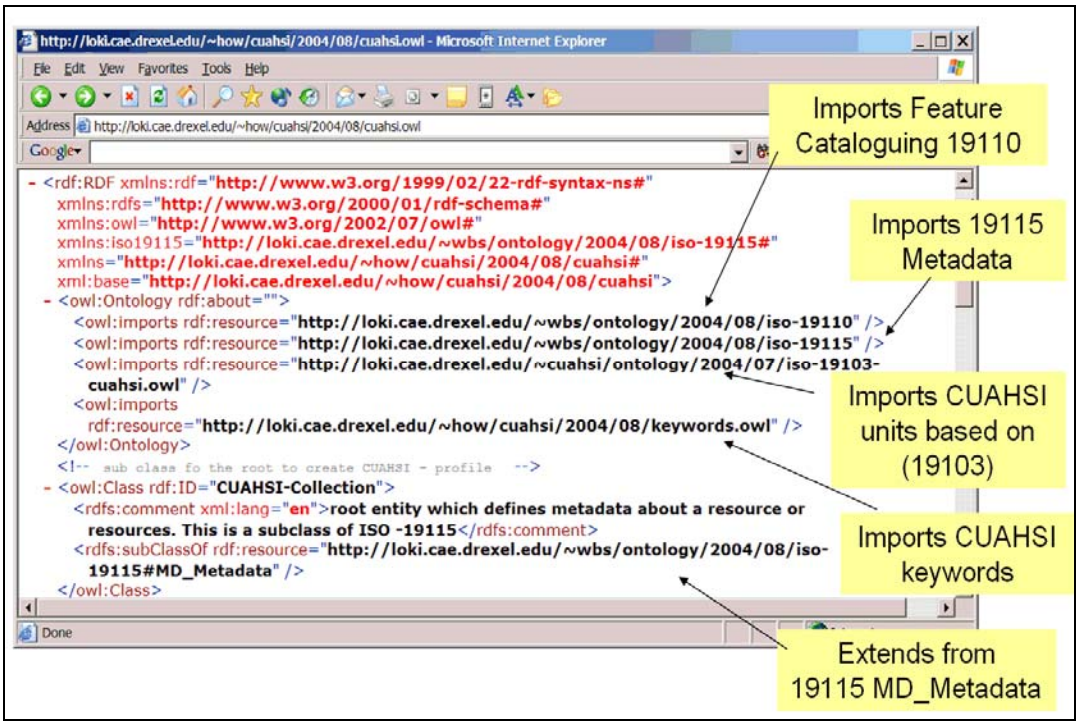


Figure 54. CUAHSI Metadata Profile

6. The elements were set as CORE using the *Pangloss* MetaExtender as shown in Figure 55. The core elements were saved in MTF (See Figure 56) and in XML as shown previously in Figure 19.

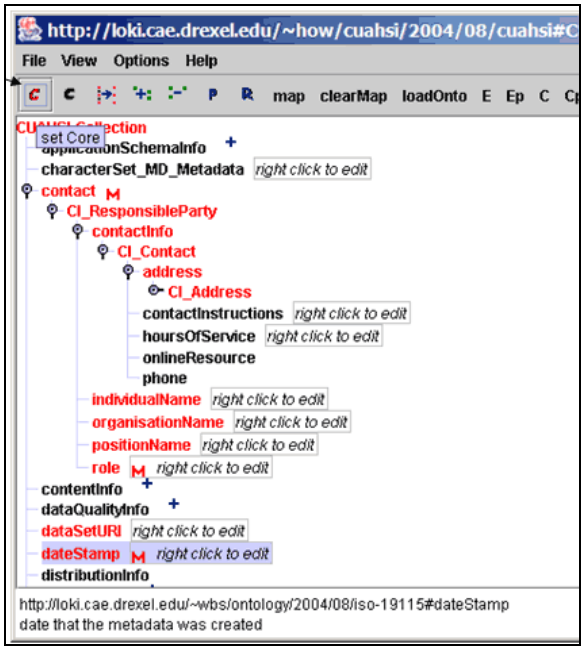


Figure 55. *Pangloss* MetaExtender snapshot of CUAHSI core

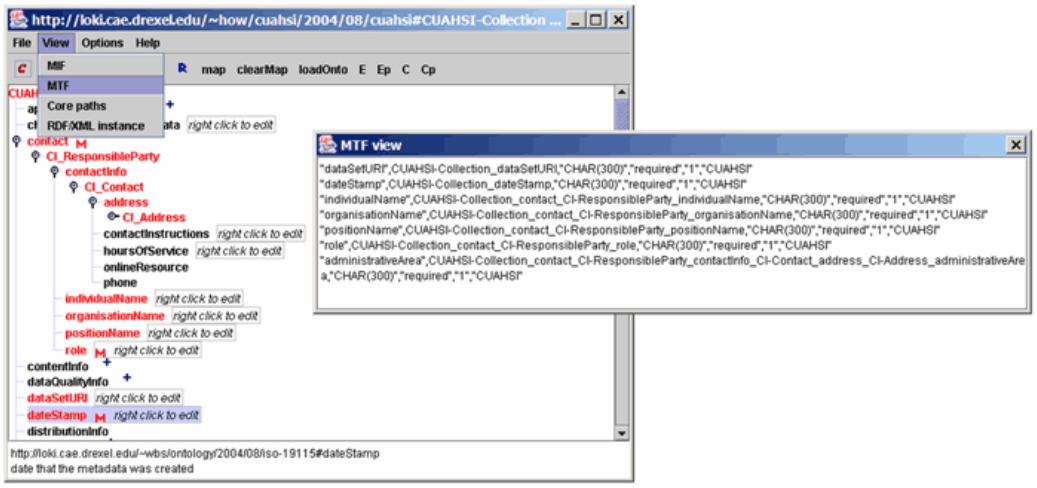


Figure 56. MTF export from *Pangloss*

The *Pangloss* MetaInstance was used to read the CUAHSI extension declaration, the controlled vocabularies and display a tree with a controlled vocabulary to allow

control over creation of metadata instances. An excerpt of the CUAHSI core elements and the control list for the keyword elements is presented in Figure 57.

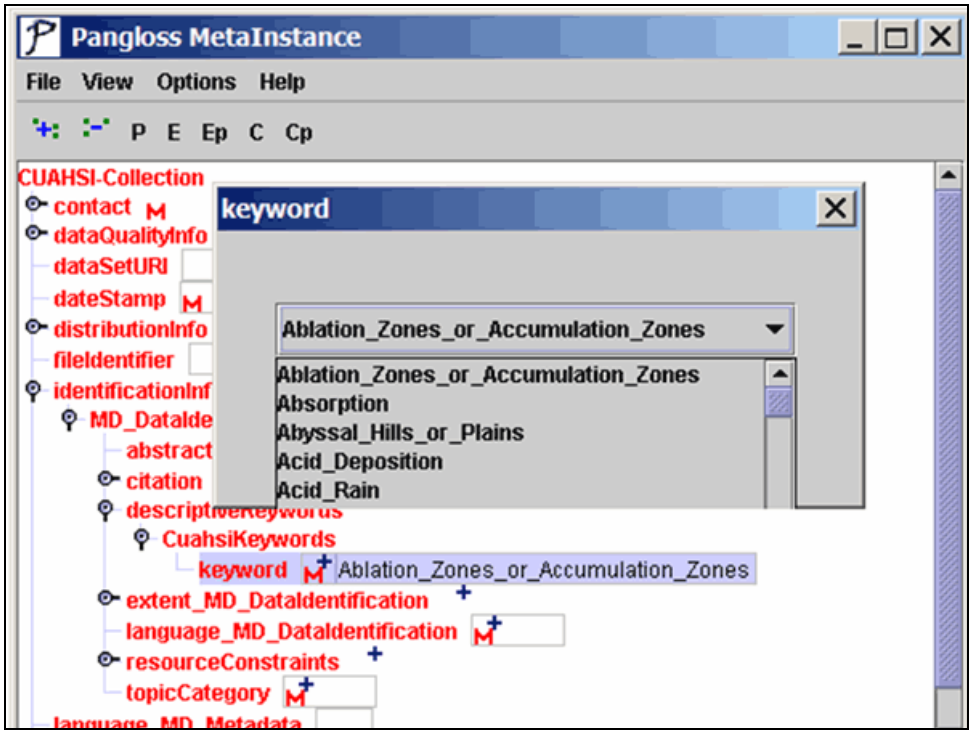


Figure 57. Controlled vocabulary of a Dynamic Community Profile.

Related ontologies for the project and more information can be found at : http://loki.cae.drexel.edu:8080/Web/how/me/metadatacuahsi.html

CHAPTER 13: INTEROPERABILITY TEST

A test was assembled to show the semantic interoperability between the Dynamic Community Profile and its parent specification. For example, if a ***profile A*** is created from the ISO specification, it will be interoperable with ISO, as well as all the other Dynamic Community Profiles directly or indirectly linked with the ISO norm (See Figure 58). The semantic interoperability means that if a query is performed using the semantics of **only** ISO, the query engine or algorithm should also be able to retrieve instances of profiles A, B, C, D..

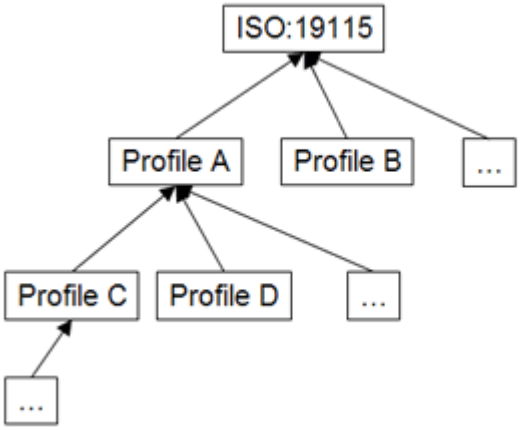
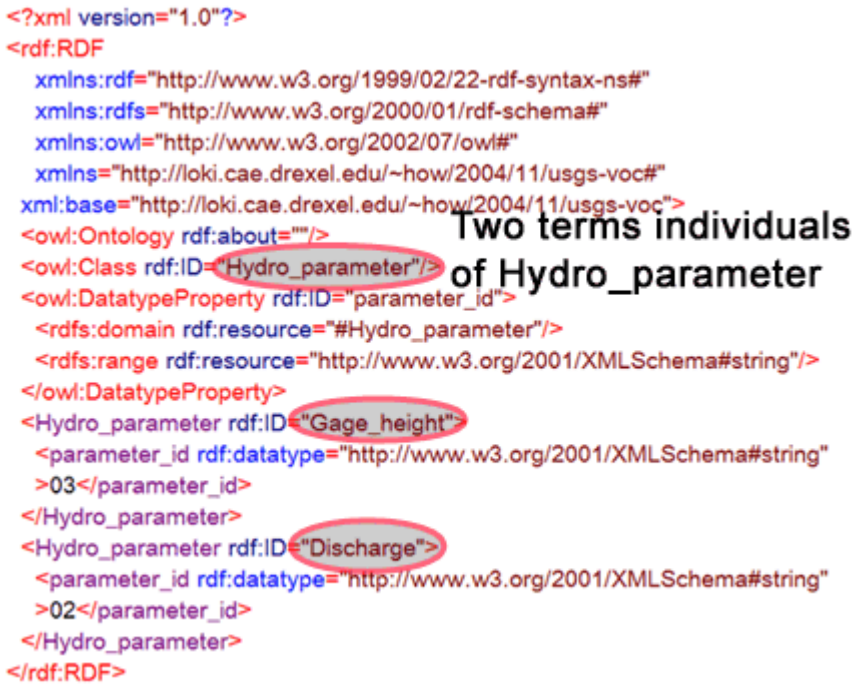


Figure 58. Profiles inheritance

The test is divided in three parts: 1) Creation of the ontologies: A controlled vocabulary; a profile and an instance; 2) defining a query, and 3) performing the query.

13.1 Creation of the ontologies

A simple controlled vocabulary was created for USGS with two terms: *Gage_height* and *Discharge*. Both of them are individuals of *Hydro_parameter.* (See Figure 59)




```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#"
  xml:base="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Hydro_parameter"/>
  <owl:DatatypeProperty rdf:ID="parameter_id">
    <rdfs:domain rdf:resource="#Hydro_parameter"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <Hydro_parameter rdf:ID="Gage_height">
    <parameter_id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >03</parameter_id>
  </Hydro_parameter>
  <Hydro_parameter rdf:ID="Discharge">
    <parameter_id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >02</parameter_id>
  </Hydro_parameter>
</rdf:RDF>
```

Figure 59. USGS vocabulary

A simple Dynamic Community Profile was created called ISO-USGS, where the resource keyword in ISO is restricted to have the USGS vocabulary shown in Figure 60.

```
<?xml version="1.0"?>

<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:iso19115="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#"
   xmlns:base = "http://loki.cae.drexel.edu/~how/2004/11/usgs-iso"
   xmlns = "http://loki.cae.drexel.edu/~how/2004/11/usgs-iso#"
>

 <owl:Ontology rdf:about="">
   <owl:imports rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115"/>
   <owl:imports rdf:resource="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc"/>
</owl:Ontology>

<owl:Class rdf:ID="USGS-Keywords">
<rdfs:subClassOf rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#MD_Keywords"/>
<rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom rdf:about="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Hydro_parameter"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#keyword"/>
      </owl:onProperty>
    </owl:Restriction>
 </rdfs:subClassOf>
```

Profile

Restrict iso keywords

Figure 60. USGS Dynamic Community Profile

An instance was created which is shown in Figure 61. The range of *descriptiveKeywords* is shown as USGS-Keywords, which is a new class that is part of the profile, and not part of the original ISO:19115. An excerpt of the instance is shown in Figure 62. Note that the class that appears as the range of *iso19115:descriptiveKeywords* is *usgs:USGS-Keywords*, but all the other resources are from the original ISO specification (i.e. elements with the prefix *iso19115*:).

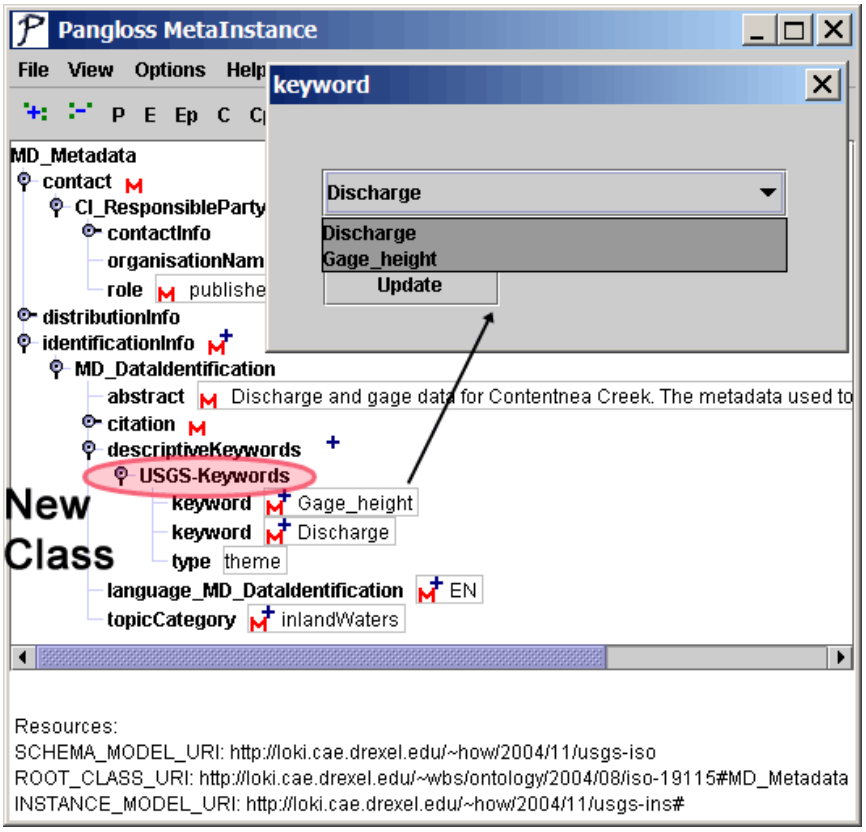


Figure 61. USGS instance creation

```
...
<iso19115:descriptiveKeywords rdf:resource="#A11595238">
  <usgs:USGS-Keywords rdf:ID="A11595238">
    <iso19115:type rdf:resource="http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#theme"/>
    <iso19115:keyword rdf:resource="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Gage_height"/>
    <iso19115:keyword rdf:resource="http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Discharge"/>
  </usgs:USGS-Keywords>
</iso19115:descriptiveKeywords>
...
```

Figure 62. Excerpt of USGS instance in XML

13.2 Query definition

A query is composed of a pair $Q = (E_q, V_q)$, where E_q is a metadata element and V_q is a value to query. In this test, both are resources identified as :

E_q=***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:keyword*** ,

V_q = ***http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Gage_height***

The result of a the query is a QueryResult, QR, that is composed of simple results SR. A Simple result is as set of pairs { $(E_1,V_1)_1$, $(E_2,V_2)_2$,…$(E_n,V_n)_n$ }. In this test n=3, for which the Es are:

E_1 = ***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:title***

E_2=***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:abstract***

E_3=***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:onLine***

A formatted result can look like Figure 63, which shows the values found for E_1, E_2 and E_3.

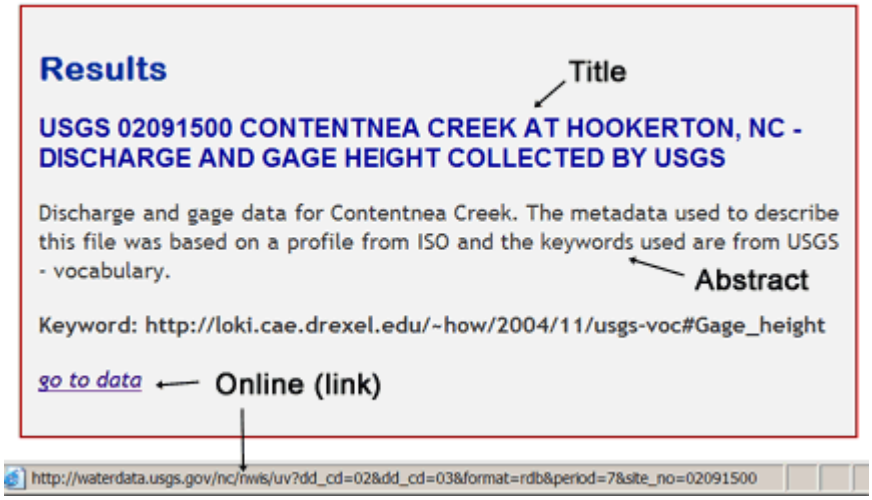


Figure 63. Formatted results

A query is executed in two phases: 1) Get the metadata root resource for all the instances whose property E_q matches V_q., and 2) starting from the metadata root resource found, get the values for E_1,E_2 and E_3.

13.3 Performing the query:

The query will be executed on the following metadata element and value:

E_q=***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:keyword*** ,

V_q = ***http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Gage_height***

To perform the query the RDF Data Query Language or RDQL is used [109]. The metadata root resource is found with the statement shown in Figure 64. The statement starts with the variable that will be returned (i.e. MD_Metadata). It has a "?" before it to denote that it is a named variable, as well as all the others that are shown with the interrogation mark. After the WHERE clause, three triples appear. The x in the first triple is the value that is going to be searched (e.i. V_q) like: ***http://loki.cae.drexel.edu/~how/2004/11/usgs-voc#Gage_height.*** The query engine will first find all the triples that match the first triple, getting all the values for *MD_Keywords*, which is used in the second triple. The query engine will then find all the triples that match the second triple finding all the possible values for *MD_DataIdentification*. Then the engine will use the *MD_DataIdentification* values to get the triples that match the third triple, and thus the values for *MD_Metadata*. The *USING* statement, helps to write the ISO resources in abbreviated form.

```
SELECT ?MD_Metadata WHERE
( ?MD_Keywords , iso:keyword , X )
( ?MD_DataIdentification, iso:descriptiveKeywords , ?MD_Keywords )
( ?MD_Metadata, iso:identificationInfo> , ?MD_DataIdentification )
USING
        iso FOR <http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#>
```

Figure 64. RDQL to find the metadata root element

The query returns the following value for MD_Metadata: ***http://loki.cae.drexel.edu/~how/2004/11/usgs-ins#A5099540***, which is shown in Figure 65.

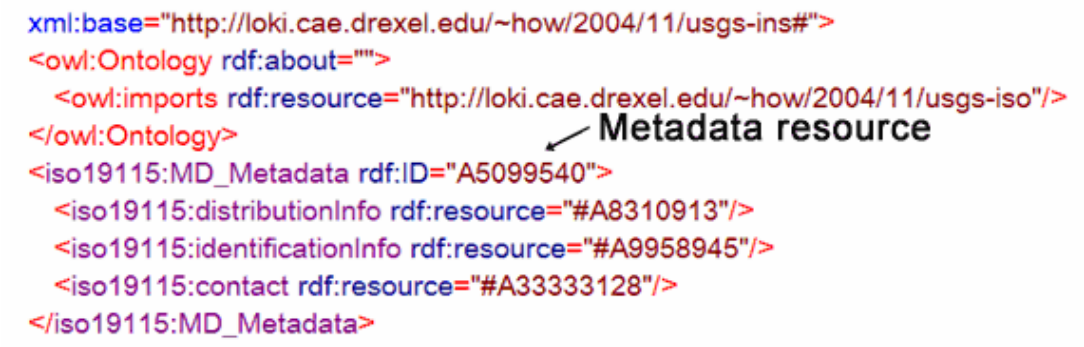


Figure 65. Instance in XML shown the metadata root

Then for the three resources E_1, E_2, and E_3 a query is performed to get the values. A query for: ***http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115:title*** is presented in Figure 66. Note that the first triple uses the value of the metadata root found. Following a similar procedure of triples pattern matching as explained previously, a value for property *title*, denoted as ?x, is found, which returns: ***"USGS 02091500 CONTENTNEA CREEK AT HOOKERTON, NC - DISCHARGE AND GAGE HEIGHT COLLECTED BY USGS".*** Similar queries can be performed to get all other values for a metadata instance.

```
SELECT  ?x WHERE
( <http://loki.cae.drexel.edu/~how/2004/11/usgs-ins#A5099540>
        , iso:identificationInfo> ,?MD_DataIdentification )
( ?MD_DataIdentification, iso:citation , ?CI_Citation )
( ?CI_Citation, iso:19115:title , ?x )
USING
        iso FOR <http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115#>
```

Figure 66. RDQL to get iso:title knowing the root value

From the previous test it can be concluded that queries are performed over properties, which are inherited by the classes, created in the Dynamic Community Profiles. The name of the classes in the profiles do not need to be known by an engine that is querying instances from the profile. Communities can create their profile, and it will be semantically interoperable with the parent metadata specification.

A complete test demo, that also shows mapping examples between metadata elements and vocabularies, can be found at http://localhost:8080/Web/how/ontomet/ontomet.jsp. The demo shows how a user searching for one term, finds metadata instances that are described with heterogeneous elements and vocabularies, as shown in Figure 67.

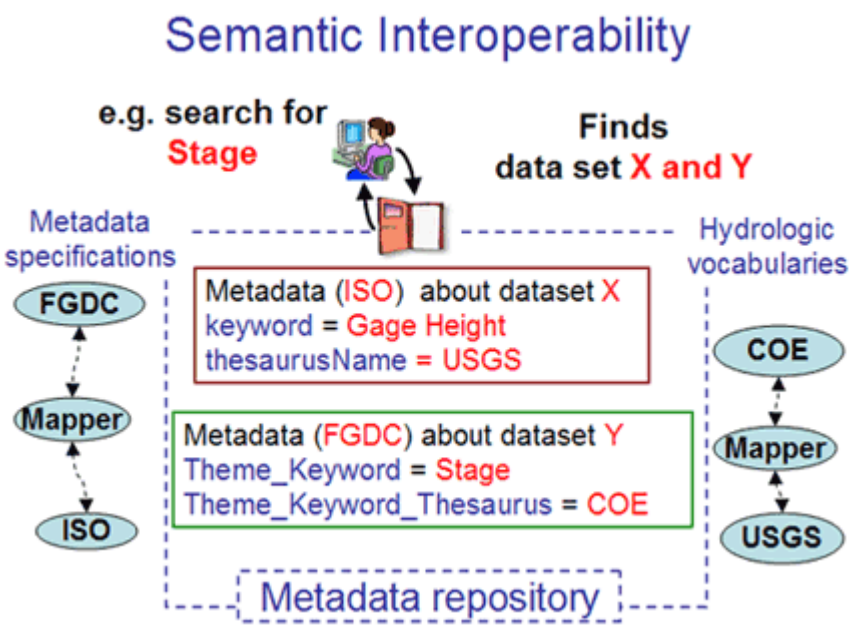


Figure 67. Semantic interoperability with mappings

The ontologies used in this demo are the following:

Specifications (Standards)

- ISO-Schema: ISO 19115-2003 and related specifications in OWL

  http://loki.cae.drexel.edu/~wbs/ontology/2004/08/iso-19115

- FGDC-Schema in OWL

  http://loki.cae.drexel.edu/~how/cuahsi/2004/10/fgdc

Vocabularies

- USGS vocabulary

  http://loki.cae.drexel.edu/~how/2004/11/usgs-voc

- COE (US Army Corps of Engineers) vocabulary

  http://loki.cae.drexel.edu/~how/2004/11/coe-voc#Hydro_Terms

Profiles

- ISO-USGS , which is a Dynamic Community Profile for USGS, where the element keywords in ISO is restricted to have USGS vocabulary
  http://loki.cae.drexel.edu/~how/2004/11/usgs-iso
- COE-FGDC, which is a dynamic profile for COE, where the element Theme_Keyword of FGDC is restricted to have values of COE vocabulary
  http://loki.cae.drexel.edu/~how/2004/11/coe-fgdc

Instances

- An instance that conforms to the ISO-USGS profile
  http://loki.cae.drexel.edu/~how/2004/11/usgs-ins
- An instance that conforms to the COE-FGDC profile
  http://loki.cae.drexel.edu/~how/2004/11/coe-ins

Mappers

- A ontology that maps USGS and COE vocabularies
  http://loki.cae.drexel.edu/~how/2004/11/coe-usgs-voc
- A mapping for the keywords element
  http://loki.cae.drexel.edu/~how/2004/11/fgdc-iso

This chapter presented a test to show the semantic interoperability between the Dynamic Community Profile and its parent specification. It was concluded that because of the queries were performed over properties, which are inherited by the classes, the Dynamic Community Profiles can be query only by knowing the semantics of its parent.

CHAPTER 14: SUMMARY

Metadata specifications can be better used by information communities if they are conceptualized, expressed in machine-readable, format and domain vocabularies can be utilized in a dynamic way. The ONTOMET framework, based on ontologies and technologies from the Semantic Web, can be used by diverse communities to achieve semantic interoperability among distributed and heterogeneous resources. The aim of this thesis was: to present an abstract model for metadata specifications, to develop a methodology to extend metadata specifications called Dynamic Community Profiles, to develop a top hydrologic ontology, to formalize semantic mappings to perform metadata crosswalks, and to provide guidelines to formalize vocabularies.

The abstract tree-path model presented in this thesis is an object model to treat metadata specifications that provides a harmonization platform to either create tools or support semantic mappings. The tool *Pangloss* was coded to successfully test this model by presenting a conceptualized specification as a tree, and allow the user to navigate along its branches. Also, representation of a metadata specification as a rooted simple tree allows creation of core paths and creation of contextual semantic mappings.

The Dynamic Community Profile methodology will allow facilitation of adopting a metadata specification, refinement of metadata elements and use of domain vocabularies in a simple declaration. Dynamic Community Profiles use the capabilities of description logics build in OWL to infer vocabularies dynamically at run time, so the latest vocabularies can be made available in the profile. Dynamic

Community Profiles will also speed up systems development times, thus avoiding to hard code the semantics.

More than one metadata specification will always exist because it is impossible to achieve a general consensus about the use of one and only one specification. To minimize the time spend in maintaining content metadata and to maximize the use among a broad range of users, metadata in one specification must be available in other related specifications. To perform metadata crosswalks, this thesis presented first, that harmonization can be achieved by expressing the source and target metadata specification in OWL; and secondly, that semantic mappings can be formalized using tree-paths. The strategy presented allowed solution of complex mappings such as one-to-many mappings.

This thesis presented strategies to conceptualize metadata specification in OWL as well as conceptualization of controlled vocabularies. ISO and FGDC conceptualizations were presented to show to different scenarios: 1) a specification conceptualized in UML, and 2) a specification not conceptualized. Controlled vocabularies for hydrology were conceptualized using a bottom-up approach starting from existing vocabularies (e.g. a list of stations of scientific term thesaurus). Also a top-down approach of a hydrologic ontology was presented, providing an upper level schema to categorize hydrologic related concepts such as features, phenomena, properties and instruments. The top hydrologic ontology presented can guide the construction of other upper domain ontologies of other Earth Science disciplines, since they share similar concepts (e.g. instrument, measurement, phenomena, etc.).

In order to support and to tie together the various ideas presented in this thesis, several stand-alone tools and Web applications were developed. These tools showed the versatility of having metadata specification, vocabularies and mapping conceptualized in a standard languages like OWL. A Dynamic Community Profile was created for CUAHSI using the previously discussed tools and methodologies. An interoperability test was also developed to show the semantic interoperability of original metadata specifications and extended ones.

CHAPTER 15: FURTHER WORK

The Semantic Web is a new technology, but there is no doubt that its adoption will spread in similar fashion as other recommended technologies by W3C have spread (e.g. HTML and XML). This thesis shows some ideas and methodologies to facilitate the use of metadata specifications using the Semantic Web; however, other strategies can be studied, where the methodologies presented in this thesis can be used. Related topics include, for example, topic maps, which is another way to express controlled vocabularies, that can be used with XPath and XML schemas technology.

Metadata instances are stored mostly in Relational Database Systems (RDBM). Conversion from triples to tuples in a relational database and vice versa is part of ongoing research, as well as translation of one query language to another (i.e. SQL and RDQL query languages). Also, automatic conversion from conceptual models to database schemas is needed to facilitate integration of metadata specification and traditional storage systems.

More ontologies for the hydrologic domain are needed. This thesis present a base, using two approaches, a bottom-up and an top-down approach. Refinement and discussion about ontology construction to promote the understating and crosswalks of vocabularies from different hydrologic communities are needed. Also development of ontologies for specific topics in hydrology, such as those related to pollutants, chemistry and water quality is a future research topic.

The model for metadata crosswalks was tested for a 1D mapping, and further development to perform a full mapping implementation is needed. The metadata mapping model approach can be combined with XSLT technologies. The

first one can be used to express the semantic mapping independently from the final format and the later to be use for the format transformations.

LIST OF REFERENCES

[1] ANSI/NISO, Guidelines for the construction, format, and management of monolingual thesauri ANSI/NISO, 2003.

[2] ANZLIC, ANZLIC Metadata Guidelines, 2001, http://www.anzlic.org.au/infrastructure_metadata.html.

[3] K. Baclawski, Kokar, M., Kogut, P., Hart, L., Smith, J., Letkowski, Jerzy, Emery, Pat, Extending the Unified Modeling Language for ontology development, Software System Model 1, 1-15, 2002.

[4] T. Baker, M. Dekkers, R. Heery, M. Patel and G. Salokhe, What Terms Does Your Metadata Use? Application Profiles as Machine-Understandable Narratives, in: International Conference on Dublin Core and Metadata Applications, K. Oyama and H. Gotoda, eds. National Institute of Informatics (NII), Tokyo, Japan, 2001.

[5] C. Batini, S. Ceri and S.B. Navathe, Conceptual Database Design, The Benjamin/Cummings publishing Company, Inc., Redwood City, California, 1992.

[6] S. Bechhofer, F.v. Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P. Patel-Schneider and L.A. Stein, OWL Web Ontology Language Reference W3C Recommendation, 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[7] Z. Bellahsene, A.B. Chaudhri, E. Rahm, M. Rys and R. Unland, Database and XML Techonoloiges, First International XML Database Symposium, XSym 2003 Berlin, Germany, September 2003 Springer, Berlin, 2003.

[8] L.E. Bermudez, Pangloss, 2004, http://loki.cae.drexel.edu:8080/~how/pangloss/.

[9] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, Scientific American 184(5), 34-43, 2001.

[10] Y. Bishr, Overcoming the semantic and other barriers to GIS interoperability, Geographic Information Science 12(4), 299-314, 1998.

[11] T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler and F. Yergeau, Extensible Markup Language (XML) 1.0 (Third Edition) W3C, 2004, http://www.w3.org/TR/2004/REC-xml-20040204/.

[12] D. Brickley and R.V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema W3C, 2004, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[13] M. Bright, A. Hurson and S. Pakzad, A taxonomy and current issues in multidatabase systems, IEEE Computer 25(3), 50-60, 1992.

[14] R. Casati and A.C. Varzi, Events, in: Stanford Encyclopedia of Philosophy (Fall 2002 Edition), E.N. Zalta, ed., 2002.

[15] CHRONOS, An Information System for Chronostratigraphy, 2004, http://www.chronos.org/index.html.

[16] J. Clark, XSL Transformations (XSLT) W3C, 1999, http://www.w3.org/TR/1999/REC-xslt-19991116.

[17] CLEANER, Collaborative Large-Scale Engineering Analysis Network for Environmental Research., 2004, http://cleaner.ce.berkeley.edu/intro.php.

[18] Commission on Geosciences Environment and Resource CGER, A Data Foundation for The National Spatial Data Infrastructure, National Academy Press, Washington, D.C., 1995.

[19] S. Cox, P. Daisey, R. Lake, C. Portele and A. Whiteside, Geography Markup Language (GML3.0) OGC 02-023r4 Open GIS Consortium, 2003, http://www.opengis.org/specs/?page=specs.

[20] CUAHSI, Consortium for the Advancement of the Hydrologic Sciences, Inc., 2004, http://www.cuahsi.org/.

[21] DCMI, Dublin Core Metadata Initiative, 2004, http://dublincore.org/.

[22] DGIWG, Digital Geographic Information Working Group, 2004, http://metadata.dgiwg.org/.

[23] DLESE, ADN framework, 2003, http://www.dlese.org/Metadata/adn-item/index.htm.

[24] Ecoinformatics, EML - Ecological Markup Language, 2003, http://www.ecoinformatics.org/tools.html.

[25] M.J. Egenhofer, Toward the Semantic Geospatial Web, in: Tenth ACM international symposium on advances in geographic information systems ACM Press, McLean, Virginia, USA, 2002.

[26] A. Elmargarmid and C. Pu, Guest Editors' Introduction to the Special Issue on Heterogeneous Databases, ACM Computing Surveys 22, 175-178, 1990.

[27] EPA, User Guide and Data Administration Guidelines for the USEPA'S Environmental Information Management System (EIMS) U.S. Environmental Protection Agency, Office of Research and Development, 2001, http://www.epa.gov/eims/EIMS_UserGuide.pdf.

[28] ESML, Earth Science Markup Language, 2004, http://esml.itsc.uah.edu/.

[29] European Territorial Management Information Infrastructure, ETeMII White Paper Chapter on Metadata, 2002.

[30] D. Fallside, C., XML Schema Part 0: Primer, W3C Recommendation W3C, 2001, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/.

[31] C. Fellbaum, Wordnet: An Electronic Lexical Database, Bradford Books, 1998.

[32] FGDC, Content Standard for Digital Geospatial Metadata, Washington, D.C., 1998.

[33] FGDC, FGDC/ISO Metadata Standard Harmonization, 2004, http://www.fgdc.gov/metadata/whatsnew/fgdciso.html.

[34] R.T. Fielding, Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, 2000.

[35] F.T. Fonseca, ONTOLOGY-DRIVEN GEOGRAPHIC INFORMATION SYSTEMS, Doctor of Philosophy, The University of Maine, 2001.

[36] F.T. Fonseca, M.J. Egenhofer, P. Agouris and G. Câmara, Using Ontologies for Integrated Geographic Information Systems, Transactions in GIS 6(3), 231 - 257, 2002.

[37] S.G. Ford and R.C. Stern, Omniport: Integrating Legacy Data into the WWW, in: Second International World Wide Web conference, Chicago, 1994, http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/CorInfSys/stern/stern-ford.html.

[38] M. Fowler, Analysis Patterns: Reusable Object Models, Addison-Welsey, Reading, MA, 1997.

[39] M. Genesereth, Knowledge Interchange Format draft proposed American national Standard (dpANS) NCITS.T2/98-004, 1998, http://logic.stanford.edu/kif/dpans.html.

[40] GERM, 2004, http://earthref.org/GERM/main.htm.

[41] GETTY, Thesaurus of Geographic Names, 2004, http://www.getty.edu/research/conducting_research/vocabularies/tgn/.

[42] Y. Gil and V. Ratnakar, TRELLIS: An Interactive Tool for Capturing Information Analysis and Decision Making, in: Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, A. Gómez-Pérez and V. Richard Benjamins, eds., Lecture in Computer Science Springer-Verlag Heidelberg, Siguenza, Spain, 2002, 2473, pp. 37 - 42.

[43] C.H. Goh, Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources, Ph.D. Thesis, MIT Sloan School of Management, 1997.

[44] J. Golbeck, G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler and B. Parsia, The National Cancer Institute's Thésaurus and Ontology, Journal of Web Semantics 1(1), 2003.

[45] J. Gomes and L. Velho, Abstraction Paradigms for Computer Graphics, The Visual Computer 11, 227-239, 1995.

[46] M. Grötschell and J. Lügger, Scientific Information Systems and Metadata Konrad-Zuse-Zentrum für Informationstechnik., Berlin, 1998, http://elib.zib.de/ftp/pub/UserHome/Luegger/Dresden/Metadata.htm.

[47] T. Gruber, A Translation Approach to Portable Ontology Specification., Knowledge Acquisition 5(2), 199-220, 1993.

[48] N. Guarino, Some Ontological Principles for Designing Upper Level Lexical Resources, in: First International Conference on Language Resources and Evaluation, Granada, Spain, 1998.

[49] N. Guarino and C. Welty, A Formal Ontology of Properties, in: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management, Lecture Notes In Computer Science, Springer-Verlag, London, UK, 2000, pp. 97-112.

[50] P. Hacker, Events, Ontology and Grammar, Philosophy(57), 477-486, 1982.

[51] T. Hadzilakos, G. Halaris, M. Kavouras, M. Kokla, G. Panopoulos, I. Paraschakis, T. Sellis, L. Tsoulos and M. Zervakis, Interoperability and Definition of a National Standard for Geospatial Data: The Case of the Hellenic Cadastre, International Journal of Applied Earth Observations and Geoinformation 2(2), 120-128, 2000.

[52] F. Harvey, W. Kuhn, H. Pundt and Y. Bishr, Semantic interoperability: A central issue for sharing geographic information, The Annals of Regional Science 33(2), 213-232, 1999.

[53] R. Heery and M. Patel, Application profiles: mixing and matching metadata schemas, Ariadne(25), 2000.

[54] D. Heimbigner and D. McLeod, A federated architecture for Information Managment, ACM Transactions on Office Information Systems 3(3), 1985.

[55] J. Helly, A.A.P. Koppers and H. Staudigel, Scalable models of data sharing in Earth sciences, Geochem. Geophys. Geosyst 4(1), 1010, doi:10.1029/2002GC000318, 2003.

[56] J. Hendler, XML and the Semantic Web, XML Journal October, 2002.

[57] G. Hodge, Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files The Digital Library Federation, Council on Library and Information Resources, Washington, DC, 2000.

[58] I. Horrocks, P. Patel-Schneider and F.v. Harmelen, From SHIQ and RDF to OWL: The Making of a Web Ontology Language, Journal of Web Semantics 1(1), 7-26, 2003.

[59] G.A. Horton, Water Words Dictionary Nevada Division of Water Resources, 2000, http://water.nv.gov/Water%20planning/dict-1/ww-index.htm.

[60] S.W. Houlding, XML an opportunity for meaningful data standards in the geosciences, Computer & Geosicences 27(7), 839-849, 2001.

[61] HP Labs Semantic Web Research, 2004, http://www.hpl.hp.com/semweb/.

[62] J. Hunter and C. Lagoze, Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles, in: The Tenth International World Wide Web Conference ACM Press, Hong Kong, 2001, pp. 457-466.

[63] IRIS, Incorporated Research Institutions for Seismology, 2004, http://www.iris.washington.edu/.

[64] A.K.M.S. Islam, L.E. Bermudez and M. Piasecki, Ontology for Geographic Information - Metadata (ISO 19115), 2004, http://loki.cae.drexel.edu/~wbs/ontology/.

[65] ISO, Geographic information - Metadata, 2003.

[66] M.N. Kamel Boulos, A.V. Roudsari and E.R. Carson, Towards a Semantic Medical Web: HelathCyberMap's Dublin Core Ontology in Protege-2000, in: Fifth International Protege Workshop, Sowerby Centre for Health Informatics at Newcastle (SCHIN). University of Newcastle upon Tyne, Newcastle, England, 2001.

[67] V. Kashyap and A. Sheth, Information Brokering Across Heterogeneous Digital Data, Kluwer Academic Publishers, Norwell, MA, 2000.

[68] M. Kifer, G. Wagner and J. Wu, Logical foundations of object-oriented and frame-based languages, Journal of the ACM 42(4), 741-843, 1995.

[69] Knowledge Information Data processing Group, Knowler:WordNet OWL-Ontology Dept. of Computer Science at the University of Neuchâtel, 2004, http://taurus.unine.ch/knowler/wordnet.html.

[70] Y. Lafon and B. Bos, Describing and retrieving photos using RDF and HTTP World Wide Web Consortium, 2002, http://www.w3.org/TR/2002/NOTE-photo-rdf-20020419/.

[71] T. Landers and R. Rosenberg, An overview of Multibase, in: 2nd International Symposium for Distributed Databases, 1982, pp. 153-183.

[72] K. Lin and B. Ludäscher, A System for Semantic Integration of Geologic Maps via Ontologies, in: Semantic Web Technologies for Searching and Retrieving Scientific Data (SCISW), Sanibel Island, Florida, 2003.

[73] W. Litwin, An overview of the multidatabase system MRDSM, in: 1985 ACM annual conference on The range of computing: mid-80's perspective: mid-80's perspective ACM Press, 1985, pp. 524-533.

[74] W. Litwin and A. Abdellatif, Multidatabase Interoperability, IEEE Computer 19(12), 1986.

[75] A. Magkanaraki, S. Alexaki, V. Christophides and D. Plexousakis, Benchmarking RDF Schemas for the Semantic Web, in: International Semantic Web Conference, Sardinia, Italy, 2002.

[76] D.R. Maidment, Handbook of Hydrology, McGraw-Hill, 1993.

[77] D.R. Maidment, Arc Hydro Gis for Water Resources, ESRI, California, 2002.

[78] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari and L. Schneider, The WonderWeb Library of Foundational Ontologies Preliminary Report WonderWeb, 2002, http://wonderweb.semanticweb.org/deliverables/documents/D17.pdf.

[79] D.L. McGuinness, Ontologies Come Age, in: Spinning the Semantic Web, D. Fensel, J. Hendler, H. Lieberman and W. Wahlster eds., The MIT Press, London, England, 2003.

[80] S. Melnik and S. Decker, A Resource Description Framework (RDF) representation of WordNet, 2004, http://www.cogsci.princeton.edu/~wn/links.shtml.

[81] P. Miller, I say what I mean, but do I mean what I say?, Adriadne(23), 2002.

[82] MMI, Marine Metadata Interoperability, 2004, http://mmi.shore.mbari.org/marinemetadata.

[83] NASA, Directory Interchange format DIF Version 9, 2004.

[84] NASA, Directory Interchange Format (DIF) Writer's Guide Global Change Master Directory. National Aeronautics and Space Administration, 2004, http://gcmd.gsfc.nasa.gov/User/difguide/difman.html.

[85] L. Neugebauer, Extending a databse to Support the Handling of Environmental Measuerement Data, in: Desing and Implementation of Large Spatial Databases, A. Buchman, G. O., J. Smith and Y.F. Wang, eds. Springer-Verlag, New York, New York, 1989, pp. 147-162.

[86] Nevada Division of Water Planning, Water Science Glossary of Terms, 2004, http://ga.water.usgs.gov/edu/dictionary.html.

[87] J. Nogueras-Iso, F.J. Zarazaga-Soria, J. Lacasta, R. Béjar and P.R. Muro-Medrano, Metadata standard interoperability: application in the geographic information domain, Computers, Environment and Urban Systems(28), 611–634, 2004.

[88] NOKIS, North and Baltic Sea Coastal Information System, 2004.

[89] NSGIC, FGDC to ISO metadata; How painful is this going to be?, in: 2003 Annual Meeting National States Geographic Information Council, Nashville, TN, 2003.

[90] NVODS, 2004, http://www.po.gso.uri.edu/tracking/vodhub/vodhubhome.html.

[91] OASIS, 2004, http://www.oasis-open.org.

[92] OCLC Online Computer Library Center, Dewey Decimal Classification (DDC), 2004, http://www.oclc.org/dewey/.

[93] OGC, Geography Markup Language (GML3.0), 2003, http://www.opengis.org/specs/?page=specs.

[94] Oklahoma Climatological Survey, Glossary, 2002, http://k12.ocs.ou.edu/teachers/glossary/g.html.

[95] L.M. Olsen, G. Major, S. Leicester, K. Shein, J. Scialdone, H. Weir, S. Ritz, C. Solomon, M. Holland, R. Bilodeau, T. Northcutt and T. Vogel, Global Change Master Directory (GCMD) Earth Science Keywords NASA, 2004, http://gcmd.gsfc.nasa.gov/Resources/valids/.

[96] OMG, Unified Modeling Language Specification, 2003, http://www.omg.org/technology/documents/formal/uml.htm.

[97] OMG, XML Metadata Interchange Specification Version 2.0, 2003, http://www.omg.org/cgi-bin/apps/doc?formal/03-05-02.pdf.

[98] OMG, Object Management Group, 2004, http://www.omg.org/.

[99] Open GIS Consortium, The OpenGIS Abstract Specification, 1999.

[100] Open GIS Consortium, Observations and Measurements, 2003.

[101] PALMM, Florida Environments Online Thesaurus, 2001, http://susdl.fcla.edu/lfnh/thesauri/feol2/index.htm.

[102] T. Ramsey, Facts and Propositions, Proceedings of the Aristotelian Society 7, 153-70, 1927.

[103] R. Raskin and M. Pan, Semantic Web for Earth and Environmental Terminology (SWEET), in: Semantic Web Technologies for Searching and Retrieving Scientific Data (SCISW), N. Ashish and C. Goble, eds., Sanibel Island, Florida, 2003.

[104] R.E. Rathmann, FGDC Metadata XML Schema 1.0.0 20030801, 2002, 2003, http://www.csc.noaa.gov/metadata/xml/tools.htm.

[105] R.B. Roberts and I.P. Goldstein, The FRL manual. AI Memo No. 409 MIT Articial Intelligence Laboratory, Cambridge, MA, 1977.

[106] W.S. Sarle, Measurement theory: Frequently asked questions, in: Disseminations of the International Statistical Applications Institute 1, ACG Press, Wichita, 1995, pp. 61-66.

[107] P. Scheuermann, C. Yu, A. Elmargarmid, H. Garcia-Molina, F. Manola, D. McLeod, A. Rosenthal and M. Templeton, Report on the workshop on heterogeneous database systems, ACM SIGMOD RECORD 19(4), 23-31, 1990.

[108] P.N. Schweitzer, Document Type Declaration (DTD) for FGDC-STD-001-1998, 2002, http://www.fgdc.gov/metadata/fgdc-std-001-1998.dtd.

[109] A. Seaborn, RDQL - A Query Language for RDF W3C, 2004, http://www.w3.org/Submission/RDQL/.

[110] SEDAC, CIESIN Indexing Vocabulary, 2004, http://sedac.ciesin.org/metadata/vocab/vocab_intro.html.

[111] A.P. Sheth, Changing focus on interoperability in information systems: from system, syntax, structures to semantics., in: Interoperating geographic information systems, M.F. Goodchild, M.J. Egenhofer, R. Fegeas and C. Cottman, eds., Kluwer Academic Publishers, Boston, 1999, pp. 5-29.

[112] A.P. Sheth and J. Larson, Federated Database Systems for managing Distributed, Heterogeneous and Autonomous Databases., ACM Computing Surveys 22(3), 1990.

[113] B. Smith and A.C. Varzi, Fiat and Bona Fide Boundaries', Philosophy and Phenomenological Research 60, 401-420, 2000.

[114] J.F. Sowa, Knowledge representation: logical, philosophical and computational foundations, Brooks/Cole Publishing Co, Pacific Grove, CA, 1999.

[115] M. St. Pierre and W.P. LaPlant, Issues in Crosswalking Content Metadata Standards National Information Standards Organization, 1998, http://www.niso.org/press/whitepapers/crsswalk.html.

[116] Standford University, Protege-2000, 2003, http://protege.stanford.edu/.

[117] S.S. Stevens, On the theory of scales of measurement., Science(103), 677-680, 1946.

[118] K. Stocks and J. Quinn, Data technologies: Geospatial data integration, in: Scalable Information Networks for the Environment (SINE). Report of an NSF-sponsored workshop, W. Michener and P. Tooby, eds., San Diego Supercomputer Center, 2002, pp. 23-29.

[119] Y. Teng, Use of XML for Web-Based Query Processing of Geospatial Data, Master Thesis, THE UNIVERSITY OF NEW BRUNSWICK, 2000.

[120] The Library of Congress, 2004, http://www.loc.gov/.

[121] The Regents of the University of California, Alexandria Digital Library Feature Type Thesaurus, 2002, http://www.alexandria.ucsb.edu/gazetteer/FeatureTypes/ver070302/index.htm.

[122] R. Thomason, Logic and Artificial Intelligence, in: The Stanford Encyclopedia of Philosophy, E.N. Zalta, ed., Fall 2003.

[123] D.S. Touretzky, The Mathematics of Inheritance Systems, Pitman Publishing Limited, London, 1986.

[124] U.S. National Library of Medicine, Introduction to MeSH in XML format,, 2001, http://www.nlm.nih.gov/mesh/xmlmesh.html.

[125] UDC Consortium, Universal Decimal Classification Consortium, 2004, http://www.udcc.org/.

[126] UKOLN, Desire Metadata Registry, 2003, http://desire.ukoln.ac.uk/registry/.

[127] UNESCO, International Glossary of Hydrology, 2002, http://webworld.unesco.org/water/ihp/db/glossary/glu/HYDR/HHYDRO.HTM .

[128] USGS, Geographic Names Information System (GNIS), 2004, http://geonames.usgs.gov/.

[129] USGS, Hydrologic Markup Language (HYDROML), 2004, http://water.usgs.gov/nwis_activities/XML/nwis_hml.htm.

[130] A.C. Varzi, Events, Truth, and Indeterminacy, The Dialogue 2, 241-264, 2002.

[131] W3C, XML Linking Language(XLink), 2004, http://www.w3.org/TR/xlink/.

[132] W3C, XML Path Language (XPath), 2004, 2004, http://www.w3.org/TR/xpath.

[133] G. Wiederhold, Mediators in the architecture of future information systems, IEEE Computer 25(3), 38-49, 1992.

[134] M. Winston, R. Chaffin and D. Herramann, A Taxonomy of Part-Whole Relations, Cognitive Science 11, 417-444, 1987.

VITA

Luis Bermudez

Education

Ph.D. Civil Engineering, Drexel University, Philadelphia, PA.	Dec. 2004
M.S. Civil Engineering, Drexel University, Philadelphia, PA.	Dec. 2001
B.S. Industrial Engineering, Andes University, Bogotá, Colombia	Sep. 1995

Award

Graduate Student Research Award, Drexel University, Philadelphia Feb. 2004

Pre-Reviewed Publications

Bermudez, L. E. and M. Piasecki (2004). Metadata Community Profiles. Submitted to Geoinformatica.

Saiful, A. S., L. Bermudez, S. Fellah, B. Beran and M. Piasecki (2004). Implementation of the Geographic Information - Metadata (ISO 19115:2003) Norm using the Web Ontology Language (OWL). Submitted to Transactions in GIS.

Conference Proceedings

Bermudez, L. E. and M. Piasecki (2004). The role of ontologies in creating hydrologic metadata. In Proceedings of The sixth International Conference on Hydro-science and Engineering, Brisbane, Australia. (In Press).

Bermudez, L. E. and M. Piasecki (2004). Achieving Semantic Interoperability with Hydrologic Ontologies for the Web. D. R. Maidment and K. Lanfear (eds.). In Proceedings of AWRA's 2004 Spring Specialty Conference Geographic Information Systems (GIS) and Water Resources III, Nashville, TN.

Bermudez, L. E. and M. Piasecki (2003). HYDROML: Conceptual development of a hydrologic markup language. In Proceedings of 30th IAHR Congress, Thessalonkiki, Greece.

Piasecki, M., L. E. Bermudez and A. S. Islam (2003). IM2: A Web-based Dissemination System for Now and Forecast Data Products. In Proceedings of 8th International Conference on Estuarine and Coastal Modeling, Monterrey, CA.